

IoT Onboarding and Proximus EnCo CloudEngine

Startup Guide

1 Resources & Prerequisites

- AllThingsTalk Rapid Development Kit
- Device UID on ATT Kit box
- ATT RDK connection schema : <http://support.sodaq.com/grove>
- Arduino IDE : <https://www.arduino.cc/en/Main/Software>
- SODAQ Mbili board support : <http://support.sodaq.com/mbili/>
 - Follow steps 2 to 5
- Install ATT SDK in Arduino IDE :
 - Download zip file from <http://docs.allthingstalk.com/developers/sdk/arduino-lorawan>
 - Under Sketch > Include library > Add .ZIP library and select the arduino-lorawan-sdk-master file
- ATT RDK :
 - Download <https://github.com/allthingstalk/arduino-lorawan-rdk>
 - Install in Arduino IDE same as for SDK above

2 Create & activate your EnCo account

Go to <http://enco.io>

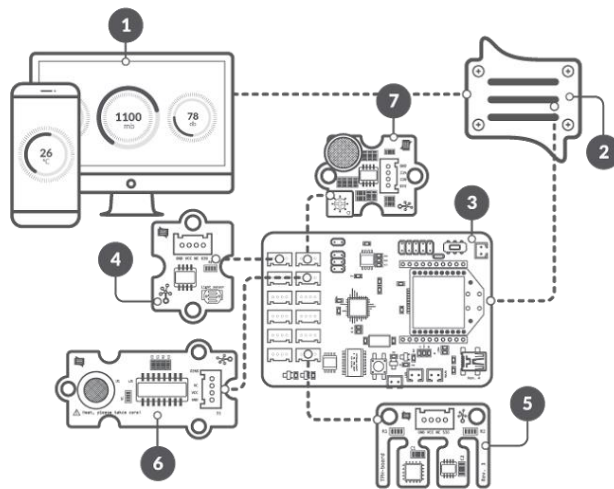
Click on register, follow the procedure.

Once registered and logged in:

- Go to EnCo Marketplace
- Select Sensor as a Service API asset, Select “Plan & Pricing”, subscribe to Welcome Plan.
Note that if your ATT kit is new (never been used before on EnCo) and you don’t plan to use other LoRa devices (other than AllThingsTalk) on your account other than this kit, you should subscribe to the Free Plan.
- If you want to send SMS from CloudEngine script, go back to EnCo Marketplace, select SMS API, select “Plan & Pricing”, subscribe to “Trial Plan”. You will need to associate a credit card to your account, this is a legal obligation needed for a complete authentication of users (anonymous SMS are not legal in Belgium). You will not be invoiced for SMS usage under the SMS Trial plan.
- Go Back to EnCo Marketplace, select CloudEngine API, register to the asset. The trial plan will allow you to run 150 free transactions. If you exceed this number, you will need to upgrade to the Premium Plan and provide your credit card details, if not associated to your account already. Please note that you will not be charged for CloudEngine transactions under the Premium plan as long as your scripts are triggered by or triggering any of the following EnCo assets:

- SMS
- Sensor-as-a-Service (SEaaS)
- Blockchain-as-a-Service
- Blockchain Management
- MyThings API
- RTCM API

3 Connect sensors and required modules to your ATT board



Attach LoRa™ module & antenna

Connect the light sensor to A2/A3 (check <http://support.sodaq.com/grove>)

Connect TPH sensor to SCL/SDA (I2C socket) (check <http://support.sodaq.com/grove>)

Connect button sensor to D20/D21 (check <http://support.sodaq.com/grove>)

Connect Sodaq Mbili to your computer through USB

4 Create your LoRa ATT device in EnCo

In EnCo, select the “DEVPORTAL”

In the Project overview, select “devices” under Sensor-as-a-Service

Click on “Activate device”, and select the “AllThingsTalk kit”

Type in the Device unique ID (label printed on the box) in the activate code field, and give your device a name.

Select your device in the list, and make sure it has the following sensors defined. If not, create these sensors. Note that once using your AllThingsTalk kit for other scenarios than the one described here, you may need to add other sensor declarations as well).

Sensors

Create a sensor

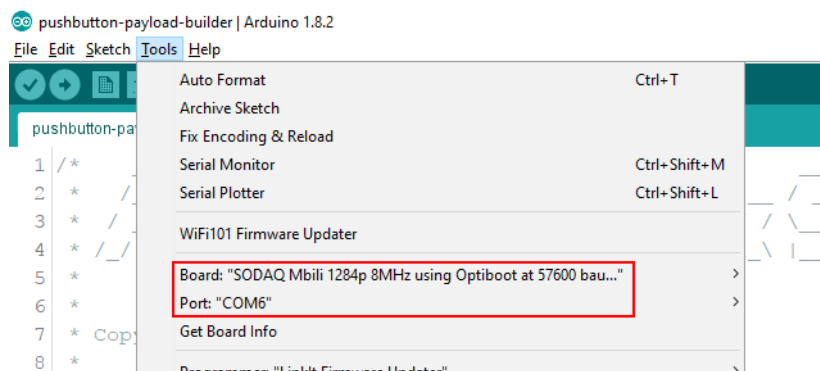


API

Sensor Name	Sensor Type	Sensor Value			
HumiditySensor	Lora_container	/	Reset Data	Delete	Edit Output
IntegerSensor	Lora_container	/	Reset Data	Delete	Edit Output
LightSensor	Lora_container	/	Reset Data	Delete	Edit Output
PressureSensor	Lora_container	/	Reset Data	Delete	Edit Output
TemperatureSensor	Lora_container	/	Reset Data	Delete	Edit Output

5 Configure your ATT board

- Connect the kit to your computer over USB
- Open your Arduino IDE
 - Go to *Tools > Board:* and select the **SODAQ Mbili** board
 - Under *Tools > Port* select the USB port to which your device is connected



- Make sure you have imported the ATT RDK from Github (see prerequisites)

We will not use the standard “environmental-sensing” sketch from AllThingsTalk. By default, the environmental-sensing sketch in the ATT RDK examples is configured to send messages every 5 minutes, and this for each sensor (T°, humidity, presence, humidity, sound, air quality), which would result in your data plan to be quickly consumed. Your SEaaS Welcome Plan allows your device to send 24 messages per day (a Free Plan for new AllThingsTalk kits is 96 message/device/day). All subsequent messages will be discarded by the network. For this workshop, we have reduced the number of sensors connected to the board to 3, and will use the button as a trigger to send the messages.

Download the Arduino schema from: <https://bit.ly/2xBt3BN>

- File : environmental-sensing-push-economic.zip
- Unzip the file on your local drive. Ensure both .ino and .h files are in the same directory

Open this ino file in Arduino IDE.

We need to add the LoRaWAN keys of our device (dev addr, apps key and nwks key) to our example sketch, so it knows where to send its data. You can find them in the EnCo DEVPORTAL, in the SEaaS asset, in your device details (keys). Fill in your keys in the keys.h file as shown below


```
uint8_t DEV_ADDR[4] = {0x08, 0x1B, 0x3A, 0x74};
uint8_t APPSKEY[16] = {0x07, 0xAC, 0x33, 0x96, 0x24, 0x72, 0xD9, 0xEB, 0xD0, 0x7A, 0xB4,
0x77, 0x75, 0xBA, 0xA8, 0xE5};
uint8_t NWKSKEY[16] = {0xC2, 0x3F, 0xB2, 0xE5, 0xDF, 0xDF, 0x2D, 0x7E, 0x2C, 0x02, 0xBE,
0xF3, 0x73, 0x1E, 0x5A, 0xE0};
```

Connect your ATT kit to your computer, check that the ATT kit's power switch is "on".

Compile the arduino sketch and check for any error (there shouldn't be any, unless you did not import both aTT SDK and RDK files correctly). Transfer the compiled sketch to your board, and open the Serial Monitor (CTRL-Shift-M). Check that the speed setting on the serial monitor is correctly set to 57600. As the compiled sketch is transferred, you should see your board come alive and initialize. The board will not send any messages unless you press the button. Don't do it now yet, you will spare some messages for later...

6 Create your flow in CloudEngine

Go to the EnCo marketplace, select the CloudEngine API, at the bottom of the page click "go to CloudEngine"



CloudEngine is the connecting hub for EnCo assets

CloudEngine's usage is free whenever a script is triggering or triggered by another EnCo asset, even for production usage. Head to our plans & pricing page for complete details.

Once registered to CloudEngine, you can access its microsite with the button below.

[Go to CloudEngine >>](#)

Or go straight to : <https://cloudengine.enco.io/>

Select "Create", give a name to your new flow.

We are going to start in graphical mode, so just click on the "+" sign in the input column. This will allow us to define on which inbound event the flow will activate. Select "SEaaS – LoRa" and close the inbound endpoint selector.

Click on the newly added LoRa inbound endpoint, and next to the "LORA" tag, add a new tag with your LORA device UID. Save and exit. We have just defined that this flow will activate for inbound LoRa messages coming from this particular device.

We will want our received messages to be send to some external HTTP endpoint. Go to <https://kara.rest/>, click on "create", then copy the Request URL. Keep this tab open, we'll need it later.

Back in CloudEngine, add an HTTP outgoing endpoint to your flow. Paste your “bin” URL from above in the URL field, make sure the method is set to POST, application/json selected as output format and type {{PAYLOAD}} in the body field. Save. Then save the flow (bottom right save button). We just stated we basically want to send out the incoming messages as we receive them.

Go back to your “bin” tab in your browser, press the button on your board. Over a few seconds time, you should see 3 separate messages coming in :

▶	POST	{ "humidity_value":46.59082,"stream_value_time":1520245772720}
▶	POST	{ "stream_value_time":1520245761593,"temperature_sensor_value":23.84}
▶	POST	{ "stream_value_time":1520245750459,"light_sensor_value":95.64117}

You can click on any message to open the twistie. Note that the stream_value_time is in epoc format. You can use an online tool such as <https://www.epochconverter.com/> to convert it to human readable format.

7 CloudEngine in script mode

Documentation about CloudEngine script mode and EnCoscript (a derivative from JavaScript) can be found here : <https://docs.enco.io/docs/script-mode>

Save your flow, then on the bottom left click on “convert to script mode”. You can see how the visual editor’s work has been translated into a script, which you are now free to adapt to achieve more advance functions. Note that once a flow has been saved in script mode, it will remain in script mode and cannot be reverted to visual mode.

For a quick introduction, you can copy/paste a sample script from <https://bit.ly/2xBt3BN>

Simply copy and paste the text of the script from the sample “ATTKit-Simple-CLOE-sample.txt” to the script editor. Before saving the script :

- Change “DEVEUI_OF_YOUR_LORA_DEVICE_GOES_HERE” to your ATT device devEUI
- Change <https://bin.dev.zlz.be/bin/8e1e8d53-b06b-4f82-8ad3-8135393d2fa0> to the URL of your paste bin created on KARA above
- Change yourmail@gmail.com to your own email address

What this script does :

- If it receives messages from your LoRa device, it will get that message payload from the received data structure, and parse its json structure
- It will dump the content of all variables to the debug windows
- It will send the complete LoRa payload to your HTTP paste bin
- If the message contains a light sensor attribute, and its value is under 40 lux, it will send a message to your email address

You can find other script samples on

- <https://github.com/Enabling/CloudengineScripts>
- <http://docs.enco.io>