UM2130

User manual

Getting started with the STM32Cube function pack for IoT node with Ethernet/Wi-Fi, NFC and sensors for vibration analysis connected with IBM Watson IoT Cloud

Introduction

FP-CLD-WATSON1 is an STM32Cube function pack. It can connect an IoT node based on the STM32L4 Discovery kit IoT node (B-L475E-IOT01A) or the NUCLEO-F429ZI to IBM Watson IoT, transmit sensor data and receive commands from remote applications. This package lets you jump-start end-to-end IoT development so that you can focus on adding desired functions.

The software includes a middleware package implementing the MQTT protocol to facilitate interaction between the featured boards and Cloud services.

The package is further extended with pre-integrated algorithms for the processing of accelerometer data which can be used to detect vibration from devices such as motors, fans and pumps. Maximum frequencies and tear/wear conditions of the device under test are reported together with raw sensor data to IBM Watson IoT thus enabling and speeding up development of solutions for industrial condition monitoring and predictive maintenance.

IBM Watson IoT parameter configuration is greatly simplified thanks to the use of NFC. The software runs on the STM32 microcontroller and includes drivers for the featured sensor devices and dynamic NFC/RFID tag.

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
AP	Access point
BSP	Base support package
FFT	Fast Fourier Transform
FOTA	Firmware update over-the-air
GPIO	General purpose input/output
HAL	Hardware abstraction layer
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
IDE	Integrated development environment
IoT	Internet of things
I^2C	Inter-integrated circuit
LwIP	Lightweight IP
MCU	Microcontroller unit
MEMS	Micro electro-mechanical systems
MQTT	Message queuing telemetry transport
NFC	Near field communication
ODE	Open development environment
REST API	Representational state tranfer APIs
RFID	Radio-frequency identification
SMD	Surface mount device
SSID	Service set identifier
UART	Universal asynchronous receiver/transmitter
URL	Uniform resource locator
Wi-Fi	Wireless lan based on ieee 802.11
WLAN	Wireless local area network

2 FP-CLD-WATSON1 software description

2.1 Overview

The software is based on the STM32CubeHAL, the hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the NFC and sensor expansion boards. The drivers abstract low-level details of the hardware and allow the middleware components and applications to access sensor data in a hardware-independent manner and to read/write information from/to NFC/RFID tag.

The package includes middleware components implementing the MQ Telemetry Transport (MQTT) application level network protocol for communication with IBM Watson IoT. MQTT is a lightweight messaging protocol with a small code footprint, low power and low bandwidth usage, particularly suitable for sensor data telemetry and implementation in embedded systems. The middleware is further extended with pre-integrated FFT algorithms to process accelerometer data by making time to frequency domain conversion.

A sample application to experiment connectivity with IBM Watson IoT platform is provided on top of the middleware stacks. Developers can use it to prototype end-to-end IoT applications, by registering the STM32 Nucleo microsystem in the IBM Watson IoT Cloud service and begin transmitting real time sensor data in a straightforward manner. Maximum frequency, maximum amplitude and alarm threshold levels are transmitted with raw sensor data to IBM Watson IoT thus enabling and speeding up development of solutions for industrial condition monitoring and predictive maintenance.

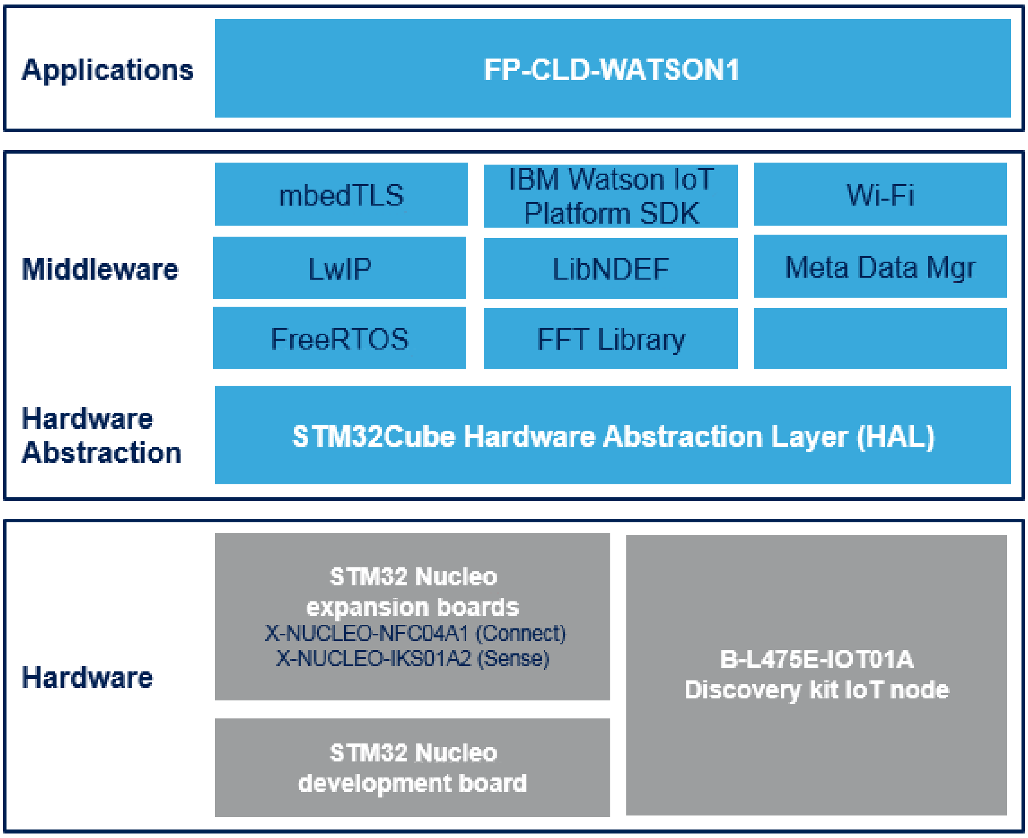
Thanks to NFC dynamic tag, you can configure Wi-Fi access point parameters and IBM Watson IoT credentials with an NFC enabled mobile phone, as well as use it to open the IBM Quickstart web page to visualize data generated by the STM32 Nucleo on-board sensors.

2.2 Features

- Complete middleware to build applications based on Wi-Fi/Ethernet connectivity, inertial and environmental sensors, and to connect an STM32 Nucleo-144 development board with STM32F429ZI MCU, or an STM32L4 Discovery kit IoT node (B-L475E-IOT01A) to IBM Watson IoT Cloud
- Software interface to access temperature and humidity sensor (HTS221), pressure sensor (LPS25HB), motion sensors (LIS3MDL, LSM303AGR, LSM6DSL) and to write and read the RFID/NFC tag (ST25DV04K)
- Integrated mbedTLS and MQTT protocol middleware
- Integrated Fast Fourier Transform (FFT) algorithm for vibration analysis
- Sample implementation based on Wi-Fi connectivity available for STM32L4 Discovery kit IoT node (B-L475E-IOT01A), based on Ethernet connectivity available for X-NUCLEO-IKS01A2, and X-NUCLEO-NFC04A1, when both connected to a NUCLEO-F429ZI
- Easy access to IBM Watson IoT Cloud services for sensors data visualization and processing (refer to http://www.ibm.com/internet-of-things/trial/ for details on license terms)
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

2.3 Architecture

Figure 1. FP-CLD-WATSON1 software architecture



The software layers used by the application to access and use the STM32 microcontroller and the sensors and NFC expansion boards are:

- **STM32Cube HAL Layer**: consists of a set of simple, generic, multi-instance APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework which allows any layers they built on, such as the middleware layer, to implement their functions without requiring specific hardware information for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability across other devices.
- **Board Support Package (BSP) Layer**: provides software support for the STM32 Nucleo board peripherals, excluding the MCU. These specific APIs provide a programming interface for certain board specific peripherals like LEDs, user buttons, etc. and can also be used to fetch individual board version information. It also provides support for initializing, configuring and reading data.

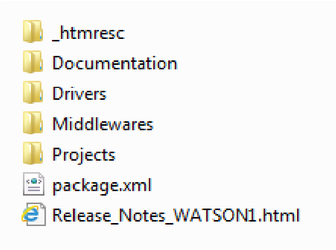
To connect the STM32 Nucleo microsystem with IBM Cloud services, we ported an open source implementation of the MQTT protocol (http://www.eclipse.org/paho/) to the STM32 platform and integrated it in STM32Cube package as a middleware library. MQTT is a lightweight messaging protocol with small code footprint, low power and low bandwidth usage. It is particularly suitable for sensors data telemetry and implementation in embedded systems (more information on MQTT protocol can be found at www.mqtt.org).

FFT middleware component computes the Discrete Fourier Transform (DFT) for a sequence of sensor output signals.

FFT is very simple and efficient algorithm and it is widely used in mathematics, science and engineering to analyze most frequencies, their amplitude and possible changes over time.

2.4 Folder structure

Figure 2. **FP-CLD-WATSON1 package folder structure**



The following folders are included in the software package:

- **_htmresc**: contains image resources used by Release_Notes.html
- **Documentation**: contains a compiled HTML file generated from the source code and documenting in details the software components and APIs.
- **Drivers**: contains the HAL drivers, the board specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS layer which is a vendor-independent hardware abstraction layer for the ARM CORTEX-M processor series.
- **Middlewares**: contains middleware interface for NFC expansion software, together with implementations for MQTT, LwIP, FreeRTOS, mbedTLS, JSON and Fast Fourier Transform (FFT) algorithms for vibration analysis.
- **Projects**: contains a sample application which uses sensor data for vibration analysis and connects with IBM Watson IoT Cloud via Ethernet when using NUCLEO-F429ZI, and via Wi-Fi when using B-L475E-IOT01. Projects are provided with three development environments (IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM), and System Workbench for STM32).

2.5 APIs

Detailed technical information fully describing the APIs functions and parameters are available to the user in a compiled HTML file in the package “Documentation” folder.

The main APIs used by the IBM sample application are:

- `int initialize_dm(char *orgId, char* domainName, char *deviceType, char *deviceId,char *authmethod, char *authToken, char *serverCertPath, int useCerts, char *rootCACertPath, char *clientCertPath, char *clientKeyPath)`: to initialize the IBM Watson IoT client.
- `int publishEvent_dm(char *eventType, char *eventFormat, unsigned char* data, enum QoS qos)`: to publish events from the device to the IBM Watson IoT service.
- `int PrepareMqttPayload(char * PayloadBuffer, int PayloadSize, char * deviceID)`: to fill the payload with the sensor values.
- `void prepare_fft_data(void); this function prepares FFT data for get_fft_data(), get_fft_max_freq() and get_fft_max_freq_amp()`
- `uint32_t get_fft_max_freq_amp(void)`: to get the amplitude of the max. frequency using Flat Top window.
- `int MQTTPublish(Client* c, const char* topicName, MQTTMessage* message)`: to publish a message to the MQTT topic defined in topicName (the result given is: success or failure).
- `uint32_t get_fft_max_freq(void)`: to get the max. frequency value.

2.6 Sample application description

A sample application is available on STM32L4 Discovery kit IoT node (B-L475E-IOT01) using Wi-Fi connectivy and on the X-NUCLEO-IKS01A2 and X-NUCLEO-NFC04A1, when both connected to NUCLEO-F429ZI, using Ethernet connectivity.

In the "Projects" directory, ready-to-build projects are available for multiple IDEs.

This application reads the data values from the temperature, humidity, pressure, accelerometer, magnetometer and gyroscope sensors and transmits them to IBM Watson IoT. When used with X-NUCLEO-IKS01A2, accelerometer data from the LSM303AGR sensor are further elaborated through FFT (feature available only with NUCLEO-F429ZI board).

The application is configured by default to run in Quickstart and Simple Registration modes by supplying the device credentials through the terminal application. When using a NUCLEO-F429ZI board, the NFC interface can also supply device credentials, as described in Configuration using NFC.

2.6.1 Vibration analysis with LSM303AGR accelerometer

Vibration analysis (VA) applied in an industrial or maintenance environment aims at reducing maintenance costs and equipment downtime by detecting equipment faults; VA is a key component of condition monitoring (CM) programs.

To analyze vibrations, accelerometer time domain signal is transferred to frequency domain. A thorough accurate analysis can be done via FFT.

Some FFT parameters are:

- FFT_max_f (Hz): indicates the maximum frequency bin of detected vibration. The accelerometer samples the signal at a selected output data rate (ODR), the default settings use 400 Hz ODR. The software calculates 512-point FFT (other options are 256 and 1024).
- FFT_max_F_amp (mg): shows the amplitude for the frequency bin. LSM303AGR accelerometer can measure very high accelerations, up to 16g.
- Motor status (OK, Warning, Failure): Three threshold levels can be set for the amplitudes. The motor condition can be monitored on the base of these acceleration limits and could be signaled.

These parameters can be easily modified in the source code. For example, more frequencies or harmonics of the signal could be monitored and added to cloud information.

By grouping adjacent frequencies, it is easier to see vibration levels arising over time. Flat-top window is used for best amplitude accuracy and different windowing functions could be created such as Hanning.

Motor status thresholds should be adjusted for each practical use case.

For best measurement accuracy, an adapter board could be mounted directly on top of DUT by creating extension cables between the sockets and connectors.

2.7 Configure IBM Watson and Wi-Fi access point parameters

Once the hardware setup is configured as described in Section 3.1 Hardware description, you need to use a serial line monitor (e.g. Tera Term) to view messages from your STM32 Nucleo based microsystem. The serial terminal must be configured with the following parameters:

- BaudRate: 115200
- Data: 8 bit
- Parity: none
- Stop: 1 bit
- Flow control: none
- NewLine RX: AUTO
- NewLine TX: CR+LF
- Local echo: Enabled

In order to connect your STM32 Nucleo based microsystem with a wi-fi access point (AP), you have to configure the correspondent SSID and password parameters. It is possible to set SSID and password by typing them into the serial interface or using NFC expansion board together with a companion mobile application.

Once parameters are set, they are written by the application in the STM32 Nucleo board flash memory and used as default values each time the board is powered on or reset. To overwrite parameters in flash, it is necessary to press the user button for some seconds when starting the firmware application.

To configure access point parameters via NFC:

1. Download on your NFC enabled mobile phone the ST25 NFC mobile application: https://play.google.com/store/apps/details?id=com.st.demo (any other mobile application able to write NDEF parameters to NFC can also be used).
2. Launch the ST25 NFC mobile app and press the “Compose NDEF” button.

2.7.1 Configuration via serial terminal

Step 1. Configure the IoT node device parameters and Wi-Fi credentials (only when using STM32L4 Discovery kit IoT node) via serial terminal

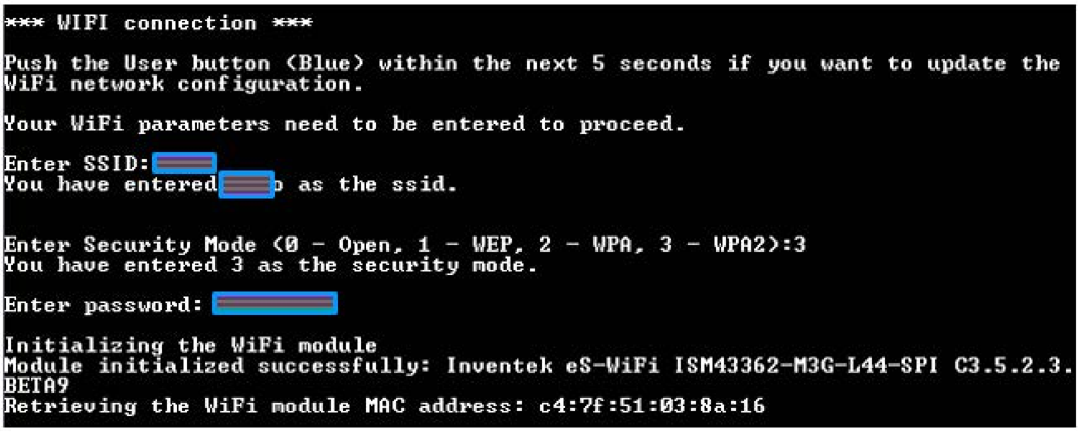


Figure 3. Configuration of Wi-Fi credentials

Step 2. As the NUCLEO-F429ZI uses Ethernet to connect to IBM Watson cloud, connect the board Ethernet port to a switch/router using an Ethernet cable

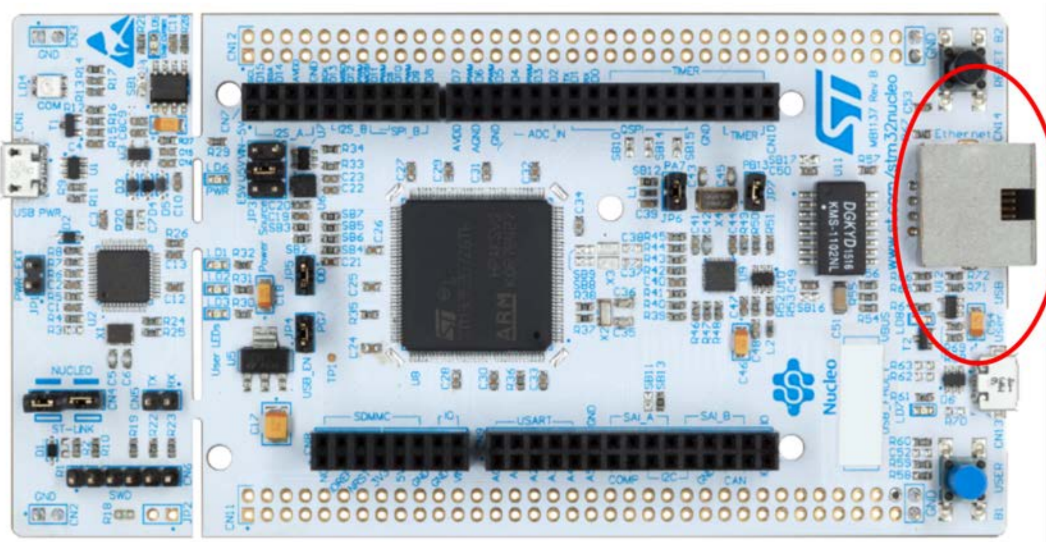


Figure 4. NUCLEO-F429ZI Ethernet port

Step 3. Enter the root CA in the serial terminal, by copying and pasting from the content of the file "Projects\Common\Bluemix\comodo_watson.pem"

Figure 5. Enter root CA certificates

```
*** Board personalization ***

*** Ethernet connection ***
Initializing LwIP on Ethernet interface

Starting DHCP client to get IP address...
IPaddress = 192.168.0.4
Mac address: 3e:1d:6d:aa:fc:0a
Retrieving the IP address.
IP address: 192.168.0.4

Updating TLS security credentials.

Enter the x509 certificates or keys as per the following format:
-----BEGIN CERTIFICATE-----
YMPGn8u67GB9t+aEMr5P+1gmIgNb1LTV+/Xjli5wwOQuvfwu7uJBVcA0Ln0kcmnL
R7EUQIN9Z/SG9jGr8XmksrUuEvmEF/Bibyc+E1ixVA0hmnM3oTDPb5Lc9un8rNsu
.......
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
YMPGn8u67GB9t+aEMr5P+1gmIgNb1LTV+/Xjli5wwOQuvfwu7uJBVcA0Ln0kcmnL
.......
-----END CERTIFICATE-----
\n.......

Enter your root CA:
```

Step 4. Select either Quickstart or Simple Registration mode

```
Enter Registration Mode (1 - Quickstart, 2 - Simple):
1
```

Step 5. Enter Configuration String

```
Enter the Bluemix connection string of your device: (template: DeviceType=xxx;D
eviceId=xxx)
DeviceType=device_type;DeviceId=id1
```

Note: This hardware configuration can be used for the setup in both quickstart and registered mode.

2.7.2 Configuration using NFC

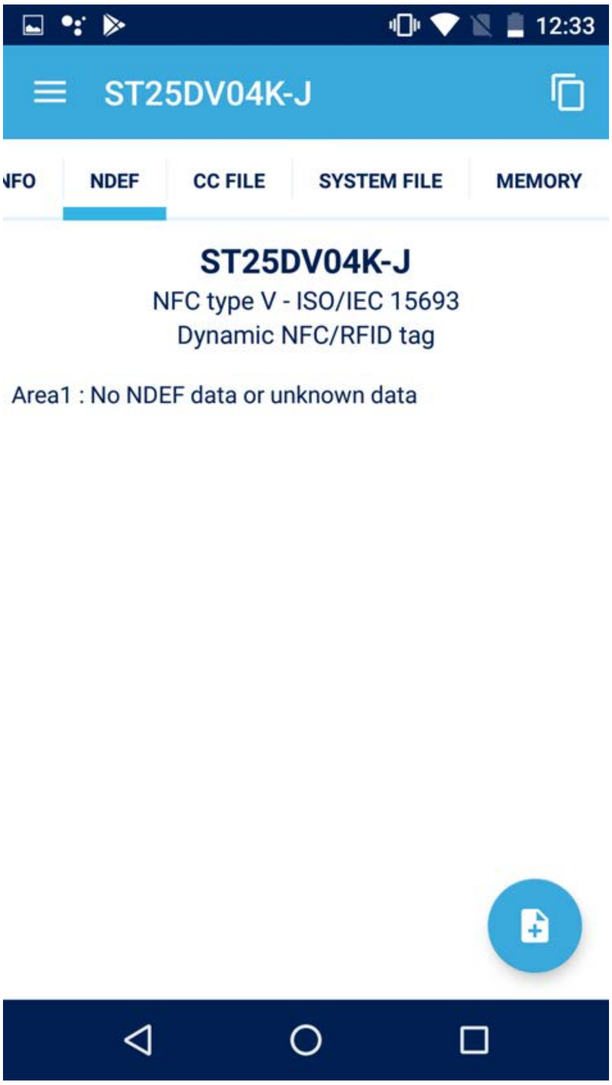
Configuration via NFC is possible only when using the NUCLEO-F429ZI board. It requires an X-NUCLEO-NFC04A1 expansion board and an Android mobile phone which supports NFC. You can configure the hardware via the ST25 NFC app.

Figure 8. ST25 NFC app for Android

Step 1. Put your Android phone close to the X-NUCLEO-NFC04A1 NFC tag.

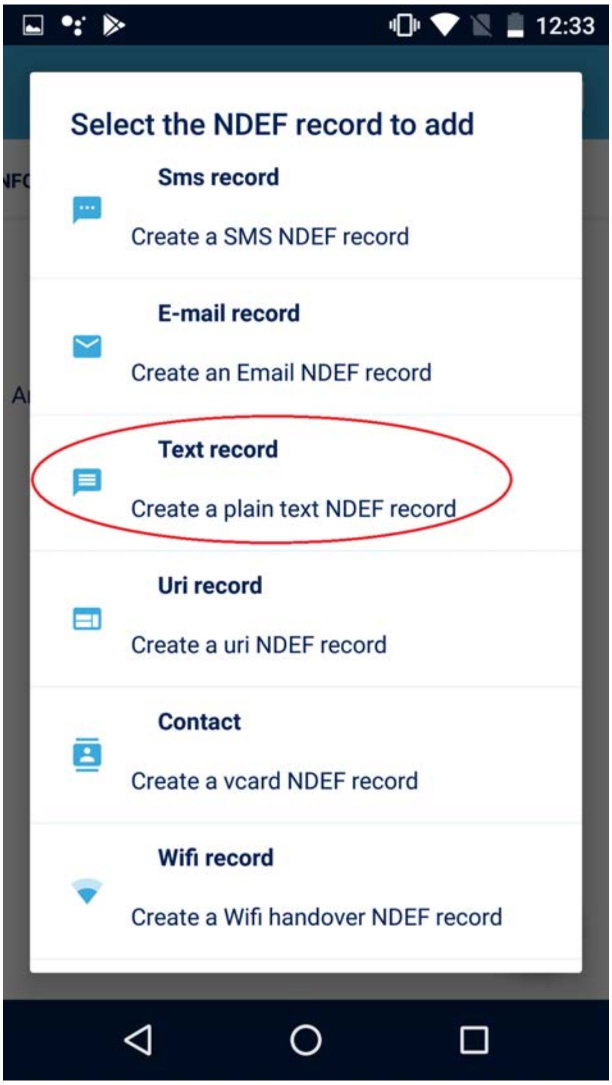
The following screen appears:

Figure 9. NDEF write menu



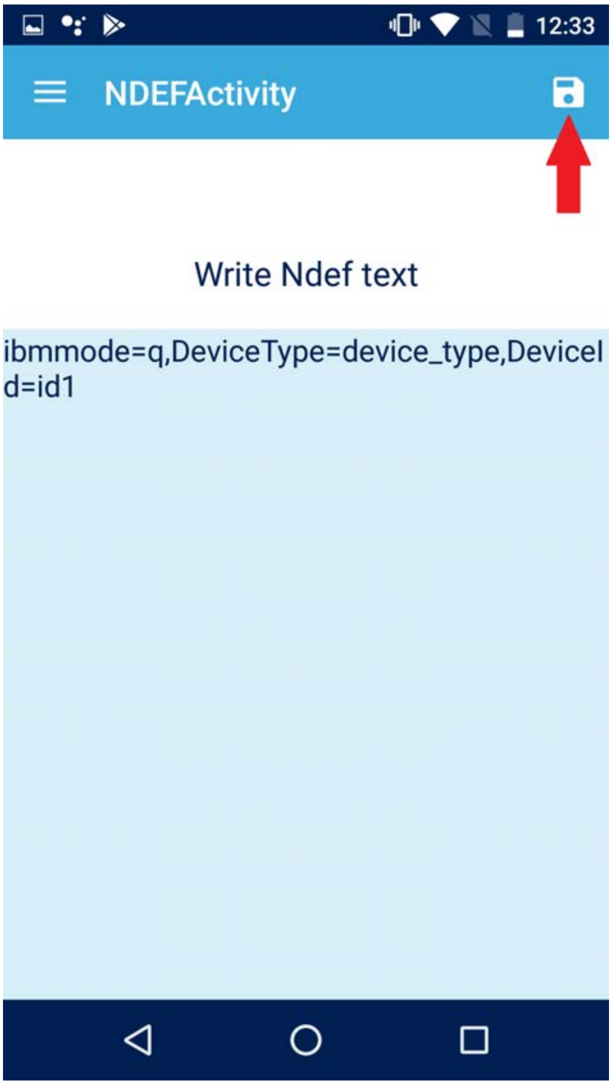
Step 2. Tap the NDEF record add button, at the bottom right of the above screenshot.
The following screen appears:

Figure 10. NDEF selection



Step 3. After selecting [**Text record**], write a formatted text string to be transferred to the NFC device by tapping on [**Save**], as shown below.

Figure 11. Write to tag



This application can be used to configure the hardware in quickstart and registered mode. The configuration parameters are given in a formatted text string like `ibmmode=q,DeviceType=your_device_type,DeviceId=id1,`, where:

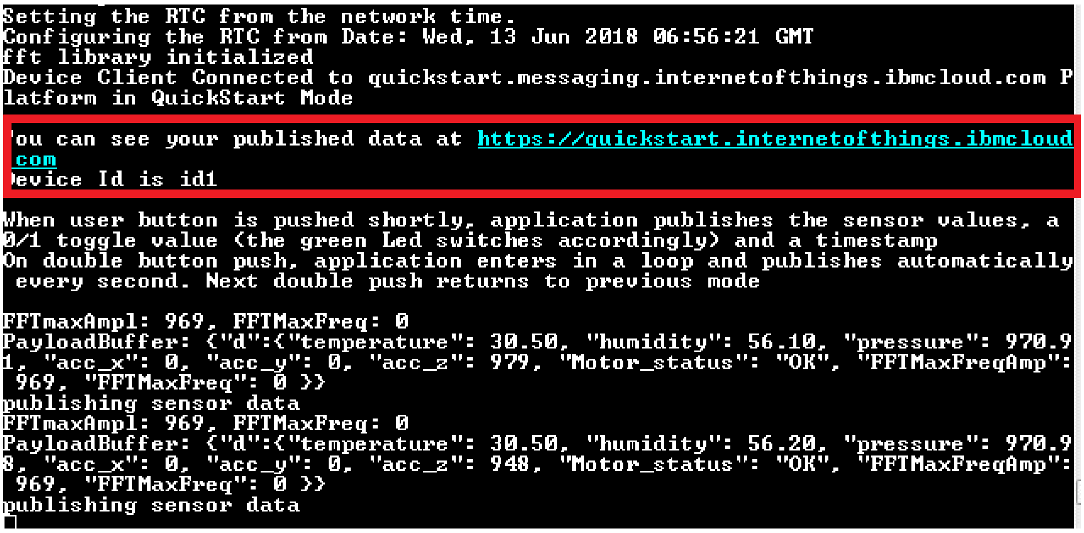
- `q` stands for "quick start mode"
- `your_device_type` is the dummy device type for your IoT node. It is not required to be registered at IBM Watson IoT platform service.
- `id1` is the dummy device ID

Note: The trailing comma has to be put at the end of the configuration string.

2.8 Viewing sensors and vibration analysis data on IBM quickstart web page

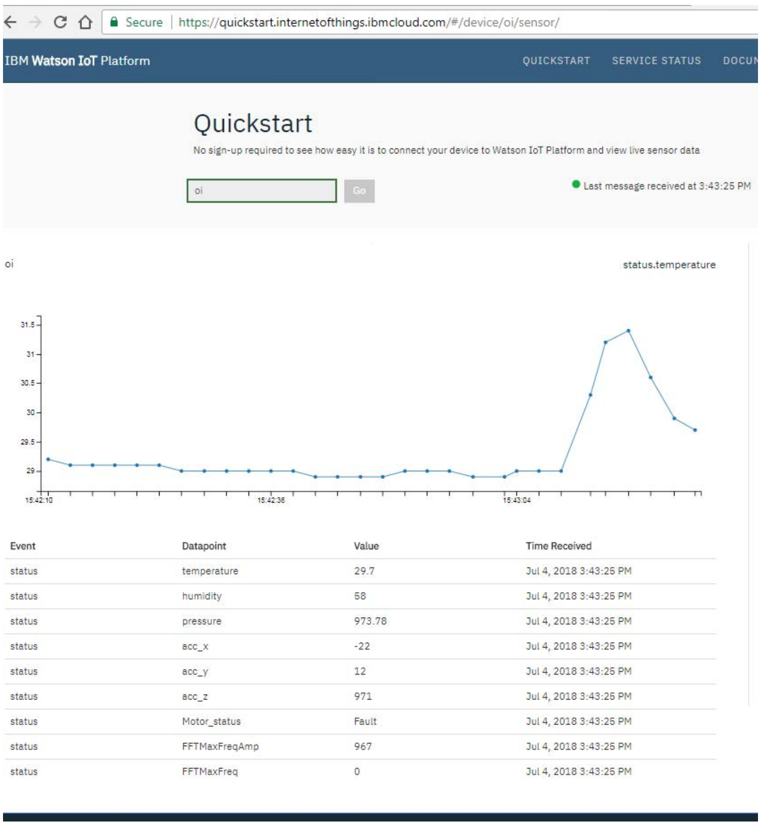
Once the IoT node configuration is complete, the console prompts for pressing the user button (blue): the IoT node, then, starts steaming sensor and FFT data to the IBM Watson IoT Cloud.

Figure 12. **Device streaming sensor data to the IBM Watson Cloud**



The device data are shown in the IBM Quickstart URL highlighted in the picture above.

Figure 13. **Sensor data shown in the IBM Quickstart web page**



If an NFC-enabled Android mobile is available, and a X-NUCLEO-NFC04A1 expansion board is mounted, you can open the Quickstart web page by approaching the phone to the NFC tag.

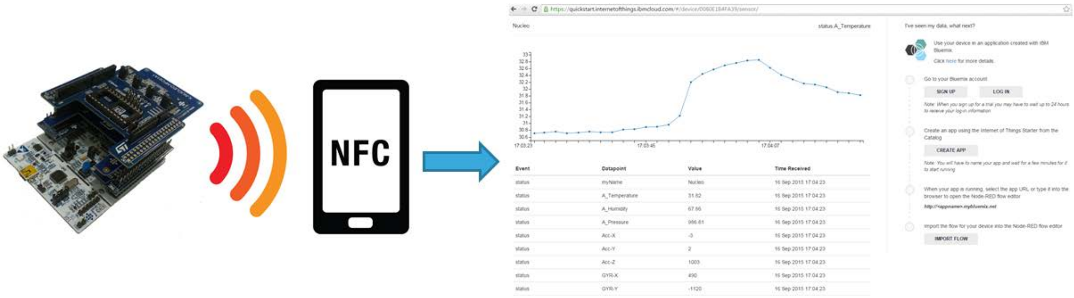


Figure 14. Automatic opening of IBM quick start web page via NFC

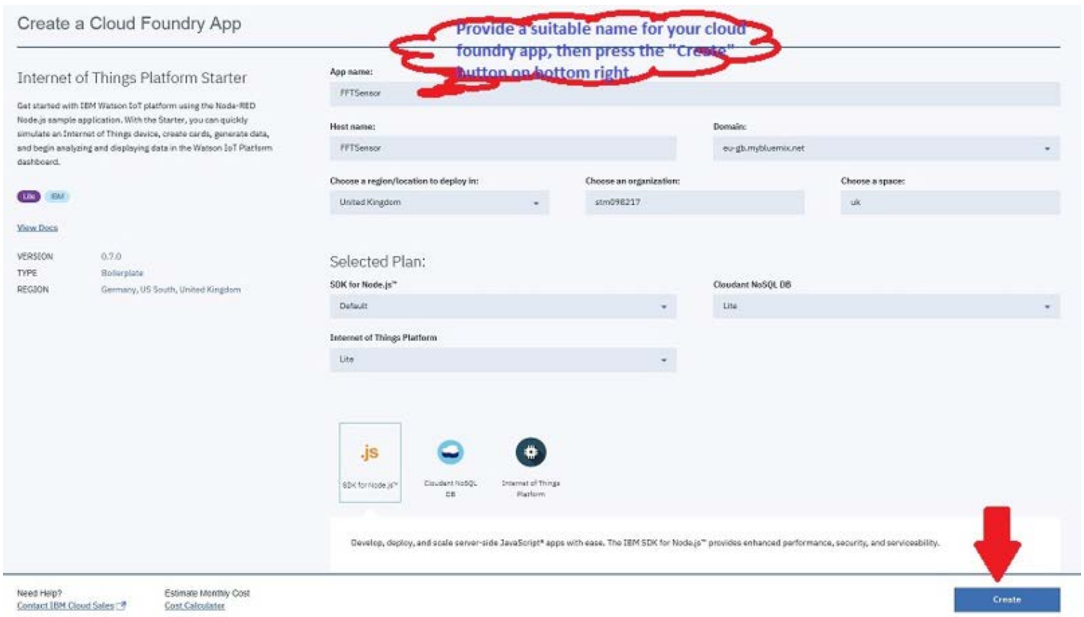
To offer a visual indication of the connectivity status, when the device is not connected to a serial terminal, the green LED2 on the STM32 Nucleo is turned ON once the board is connected to IBM Watson IoT: it blinks each time a sample of sensor data is transmitted.

2.9 Connect to IBM Watson IoT Cloud as a registered device

Registered mode allows connecting your STM32 Nucleo and expansion boards to IBM Watson IoT Cloud and build scalable IoT applications. To register the STM32-based microsystem to IBM Watson IoT Cloud, you must create an account in IBM Watson Cloud following the instructions provided at https://www.ibm.com/marketplace/cloud/internet-of-things-cloud/us/en-us.

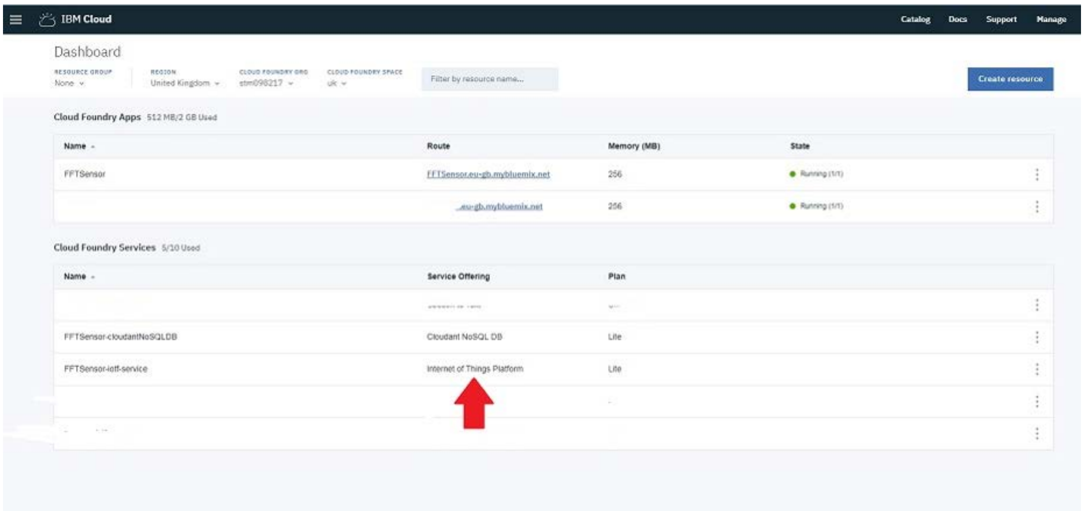
Step 1. Once signed in, use the Internet of Things Platform Starter boilerplate to create a **Cloud Foundry App** as shown below.

Figure 15. Internet of Things Platform Starter Boilerplate: Cloud Foundry App creation



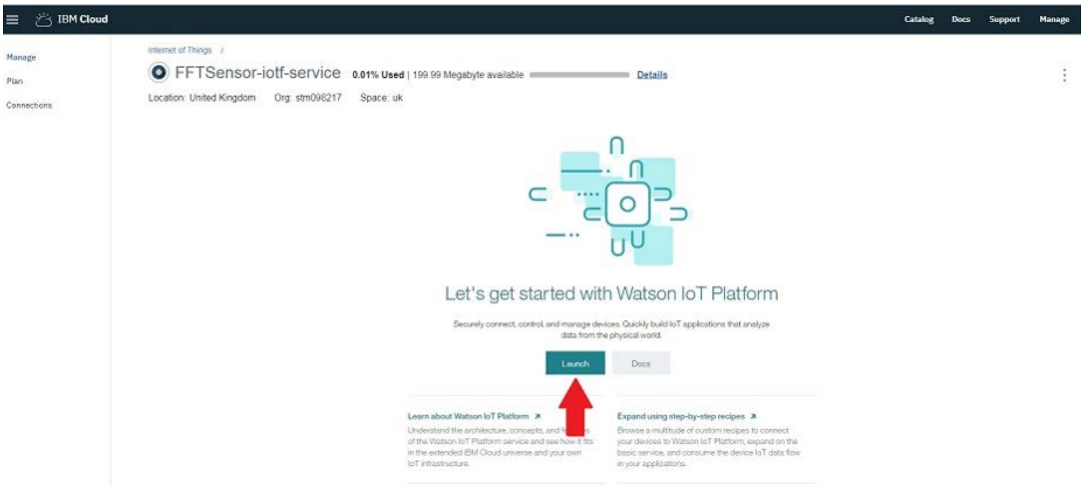
Step 2. Launch Watson IoT Platform service by clicking on the xxxx-iotf-service link in your IBM dashboard.

Figure 16. Watson IoT Platform service



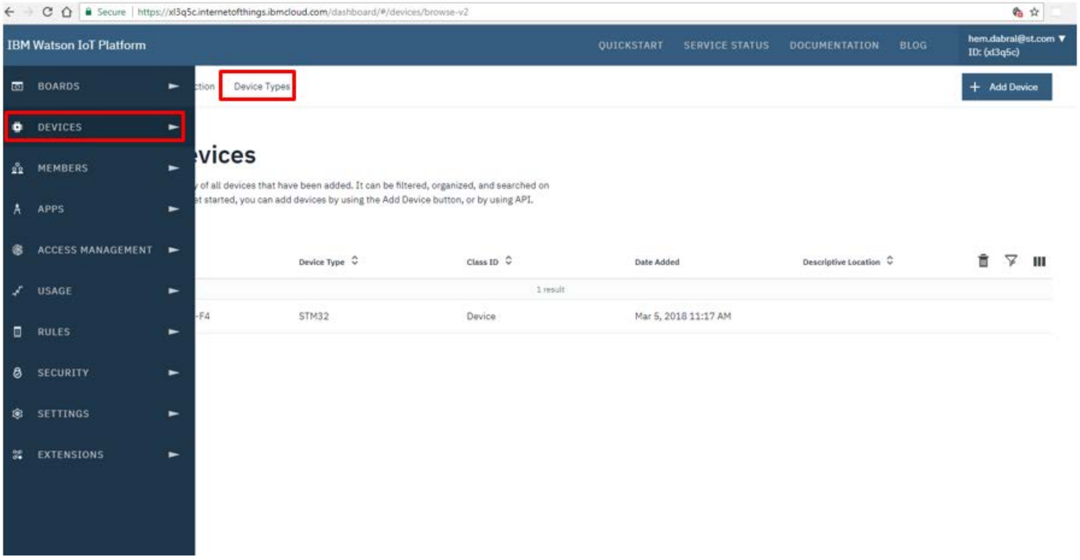
Step 3. Click on [**Launch**] to open IBM Watson IoT platform service.

Figure 17. Internet of Things Platform Starter Boilerplate: launching Watson IoT Platform



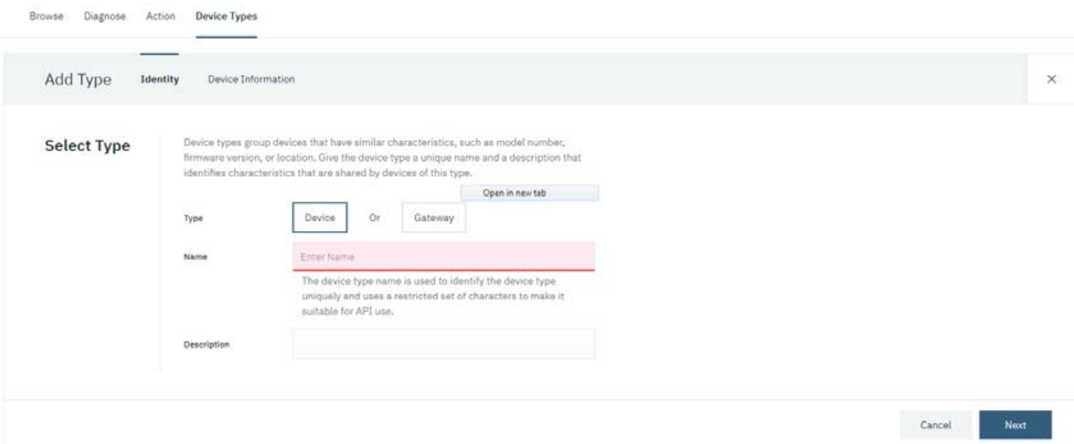
Step 4. Select [**Device**], then [**Device Type**].

Figure 18. **Internet of Things Platform Starter Boilerplate: device type**



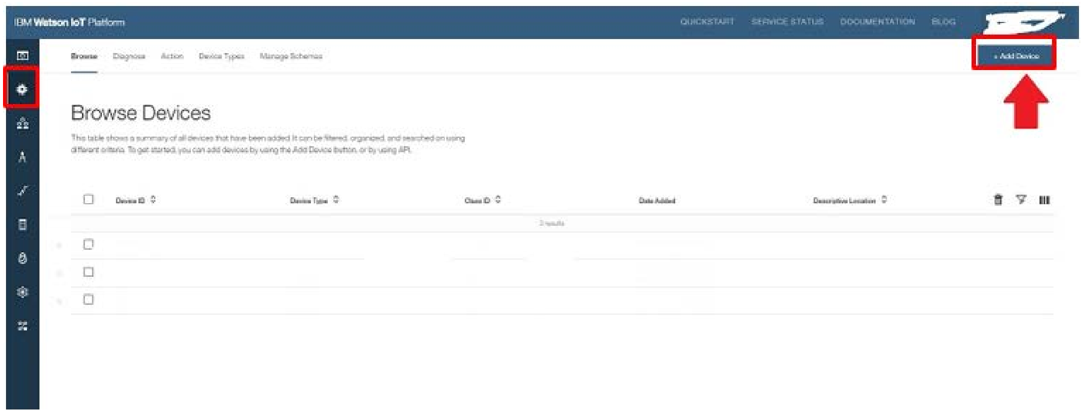
Step 5. Enter [**Device Type name**] then click on [**Next**]

Figure 19. **Internet of Things Platform Starter Boilerplate: device name**



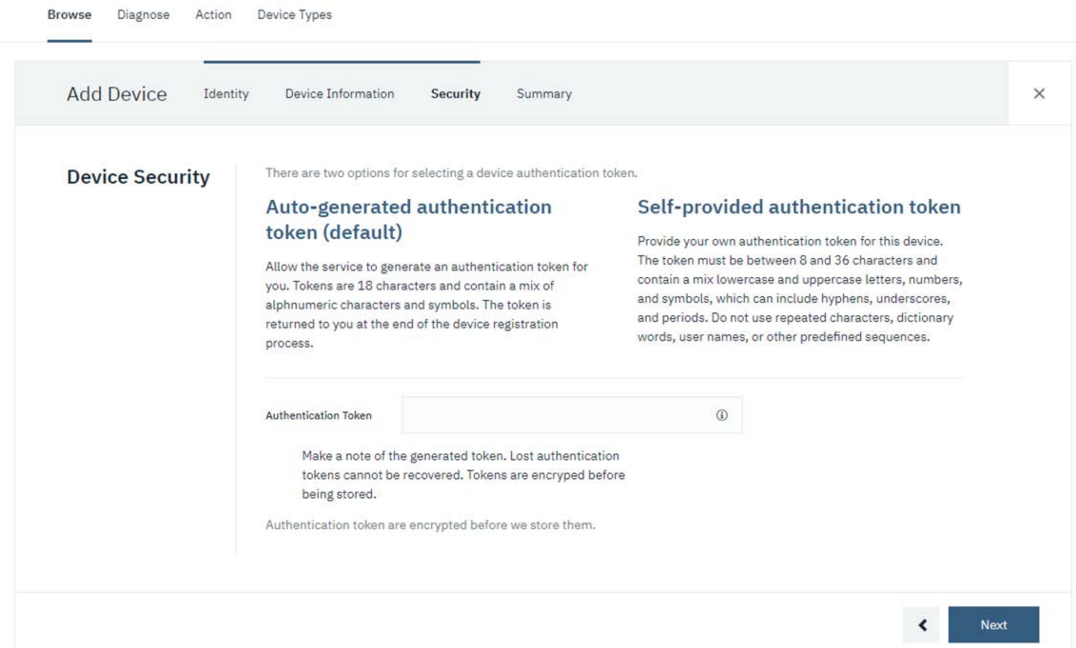
Step 6. Select [**Browse Devices**] in the left side bar, then click on [**Add Device**]

Figure 20. **Internet of Things Platform Starter Boilerplate: add device**



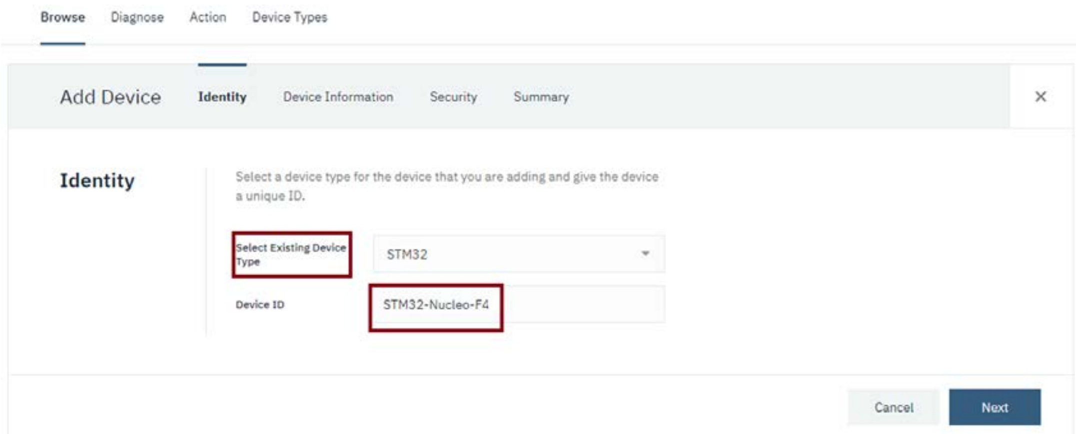
Step 7. Select [**Auto-generated authentication token**], then click on [**Next**]

Figure 21. **Internet of Things Platform Starter Boilerplate: auto-generated token**



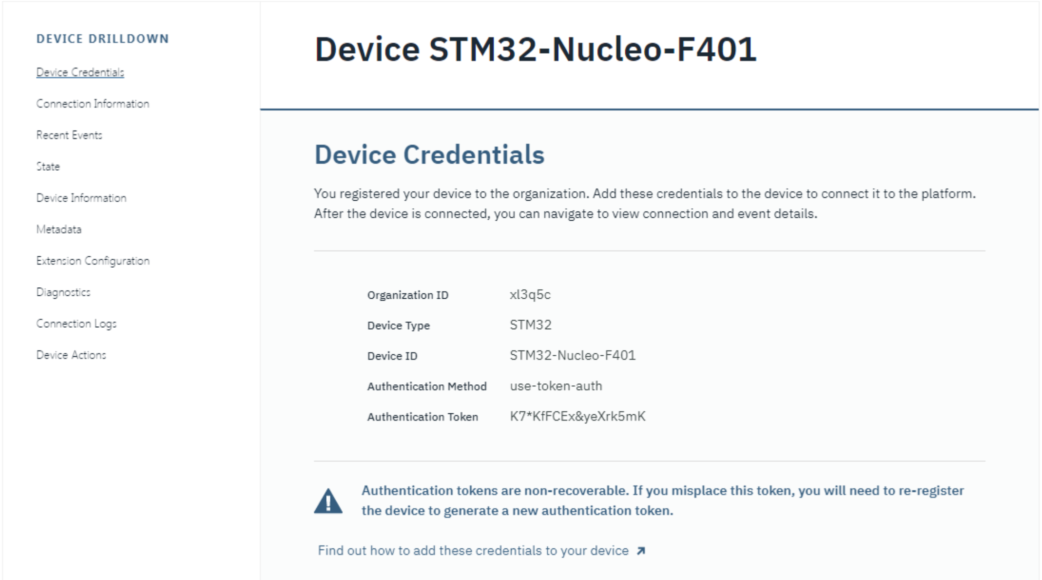
Step 8. Select [**Device Type**], enter your [**Device ID**], then click on [**Next**]

Figure 22. **Internet of Things Platform Starter Boilerplate device ID**



Note: Write down the device credentials generated as they are necessary to configure your IoT node via terminal or NFC (available for NUCLEO-F429ZI only).

Figure 23. **Internet of Things Platform Starter Boilerplate device credentials**

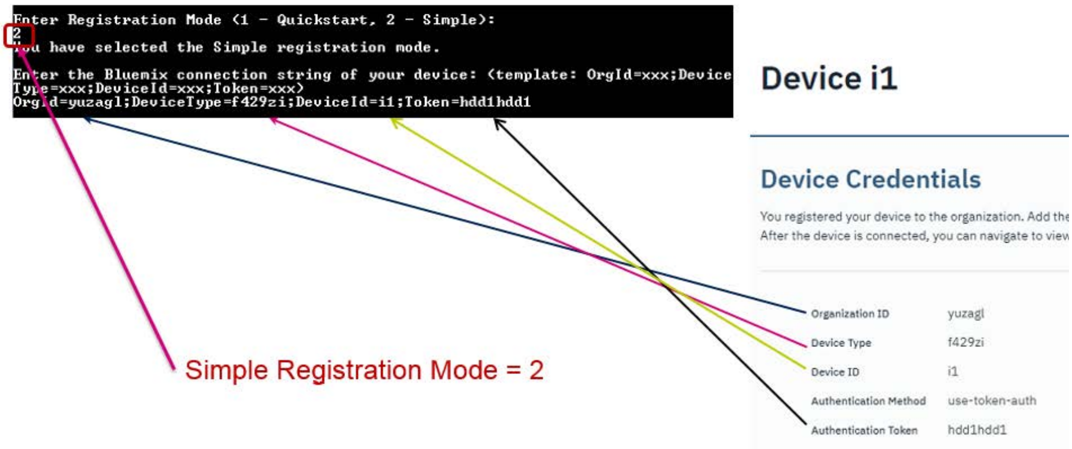


Step 9. Reboot the STM32 Nucleo board.

Step 10. When requested, select [**Registered mode**].

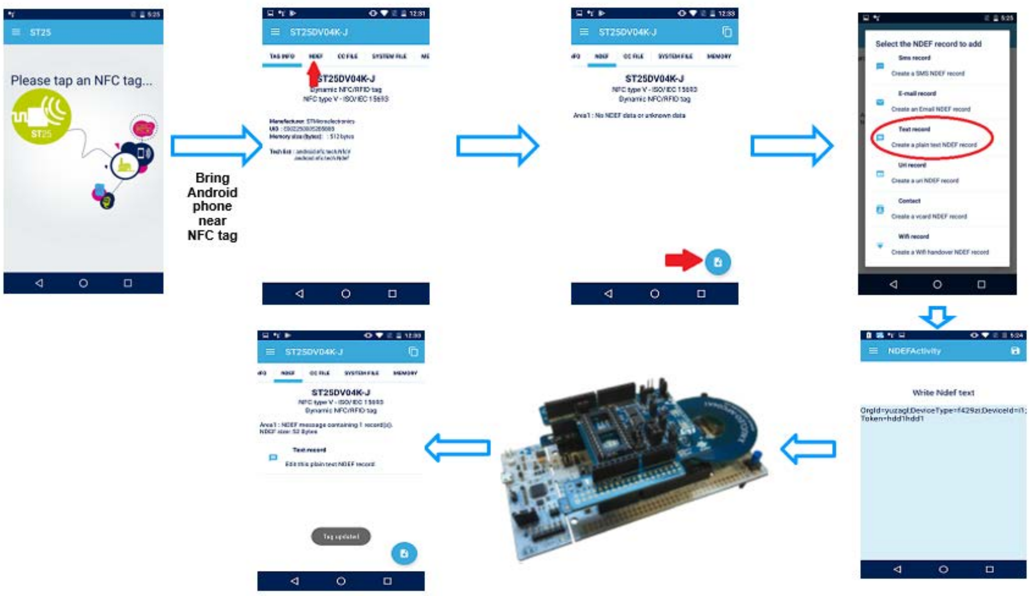
Step 11. Enter device credentials.

Figure 24. **Internet of Things Platform Starter Boilerplate registered mode**



Note: Device credentials can also be trasnsmitted via NFC through an Android mobile application (ST25 NFC app) on the basis of the workflow shown below.

Figure 25. **Registration mode device parameter provisioning using ST25 NFC app**

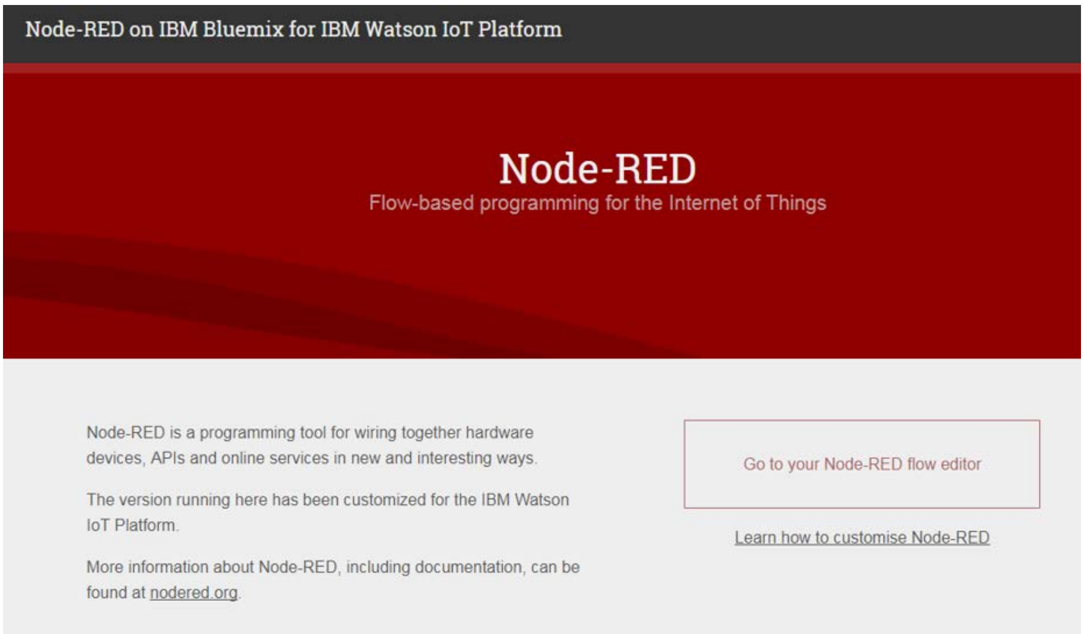


Once registered and connected to the IBM Cloud, the STM32-based microsystem is able to send and receive information to and from IBM Cloud applications and IBM Watson IoT (for further information, refer to https://www.ng.bluemix.net/docs/starters/Node-RED/nodered.html).

2.10 Connect to Node-RED

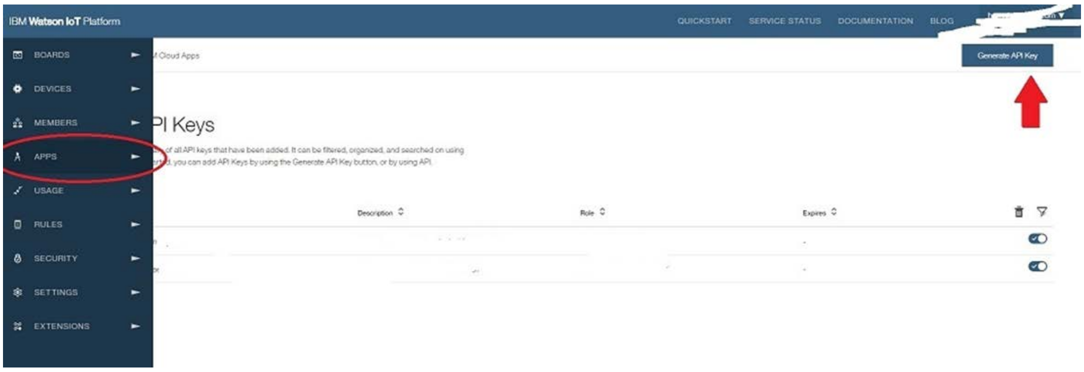
Node-RED is a flow-based development tool to link hardware devices, APIs and online services (nodered.org). It is integrated in the IBM Watson IoT Platform.

Figure 26. **Node-RED flow editor for IBM Watson IoT Platform**



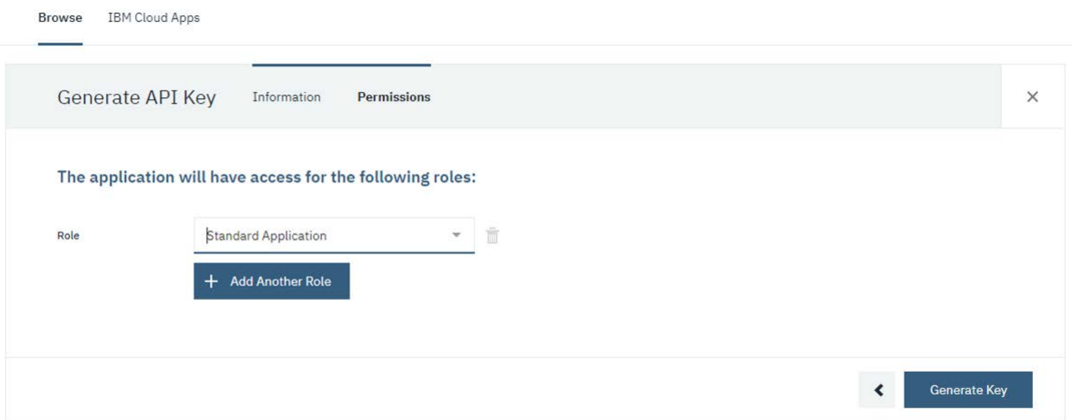
Step 1. To use the Node-RED flow editor, before connecting your devices to Node-RED, generate the API keys in the IBM Watson IoT dashboard as shown below.

Figure 27. **Generate API Key and Authentication Token (1 of 3)**



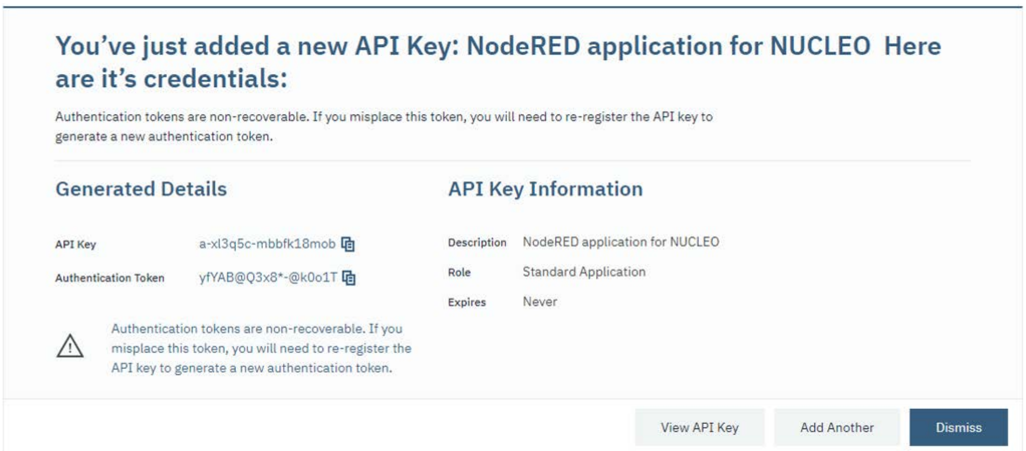
Step 2. Select [**Standard Application**], then click on [**Generate Key**]

Figure 28. **Generate API Key and Authentication Token (2 of 3)**



Take note of the API Key and Authentication token for future usage

Figure 29. **Generate API Key and Authentication Token (3 of 3)**

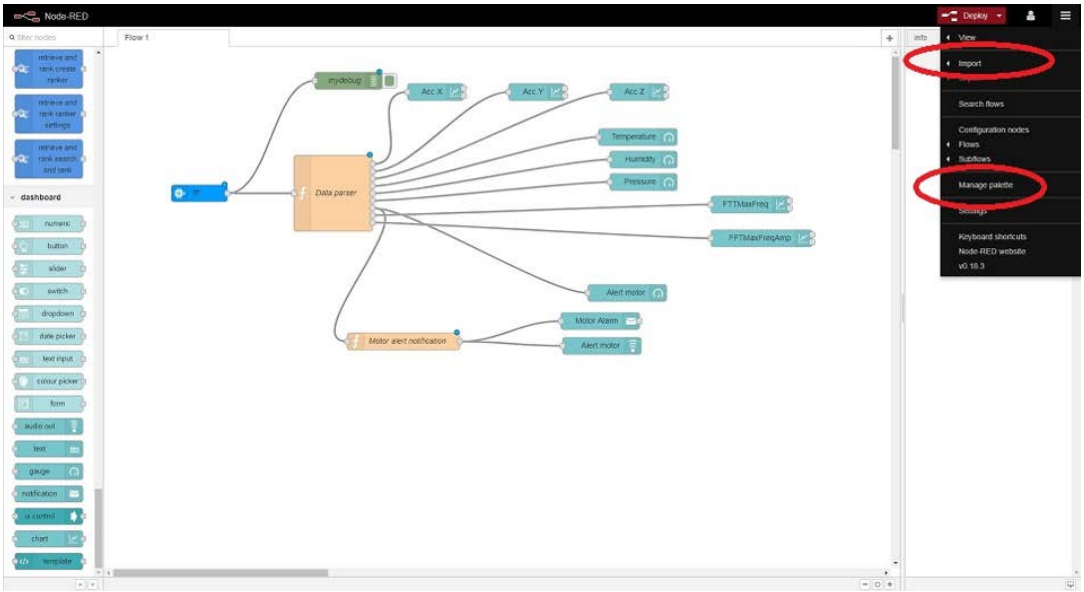


Step 3. Go back to your IBM Cloud dashboard and click on your app URL to open Node-RED flow editor. You can secure the connection by using username and password.

Figure 30. Launching Node-RED flow editor

Step 4. In Node-RED flow editor, select [**Manage Palette**] from menu option, click on the [**Install**] tab and add the Node-RED dashboard to your Node-RED palette as shown below.

Figure 31. Connect device to Node-RED application



Step 5. Click on [**Import**] to import the flow described in the file FFTSensorFlow.json (in the folder STM32CubeFunctionPack_WATSON1_F4_Vx.y.z/Utilities/NodeRED).

Step 6. Configure the Watson IoT platform node using your IoT node device parameters (Device Type, Device Id, API key and Authentication token)

Figure 32. **Configuring IoT node device parameters (1 of 2)**

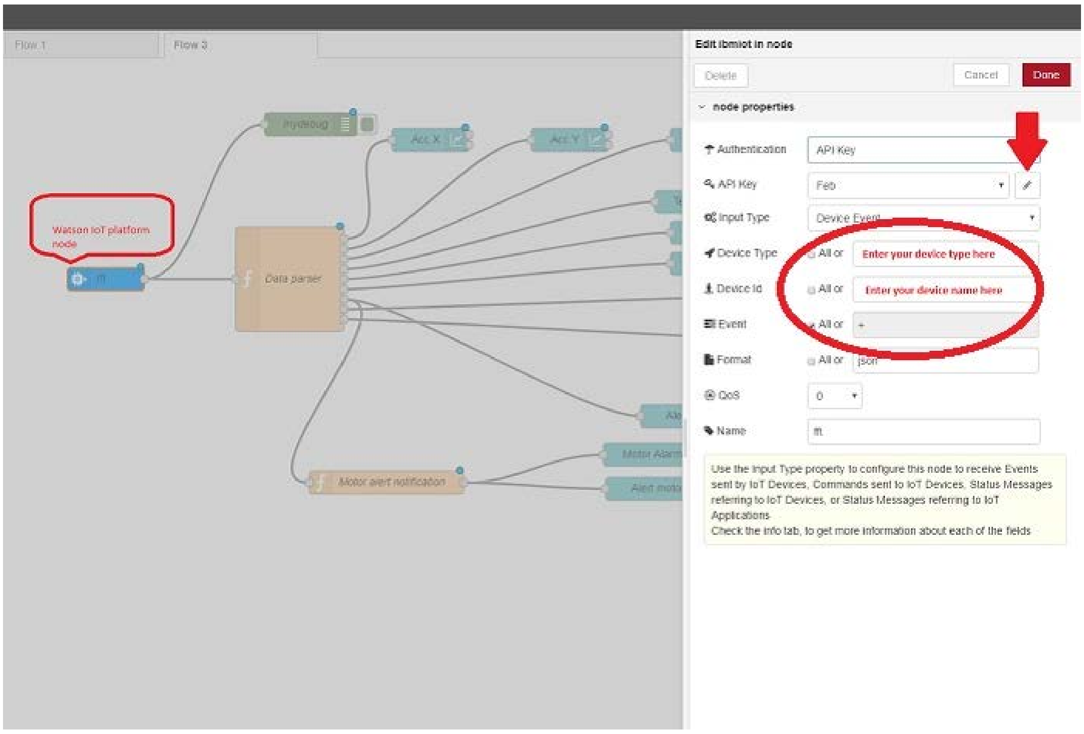
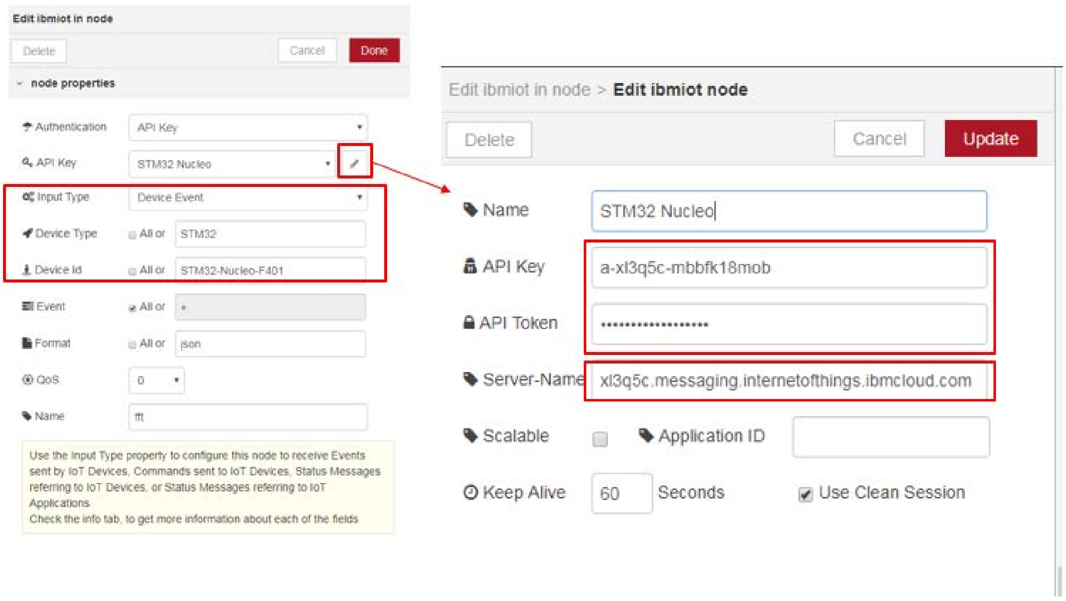
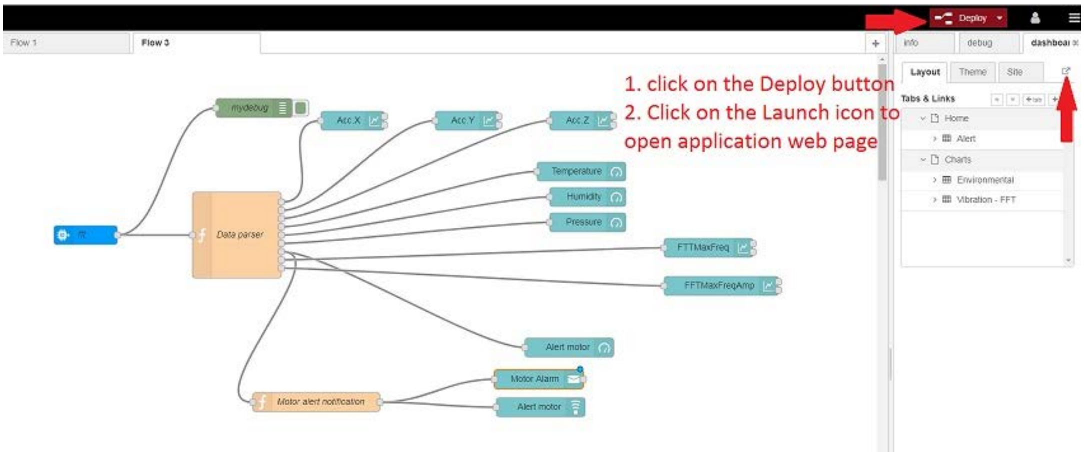


Figure 33. **Configuring IoT node device parameters (2 of 2)**



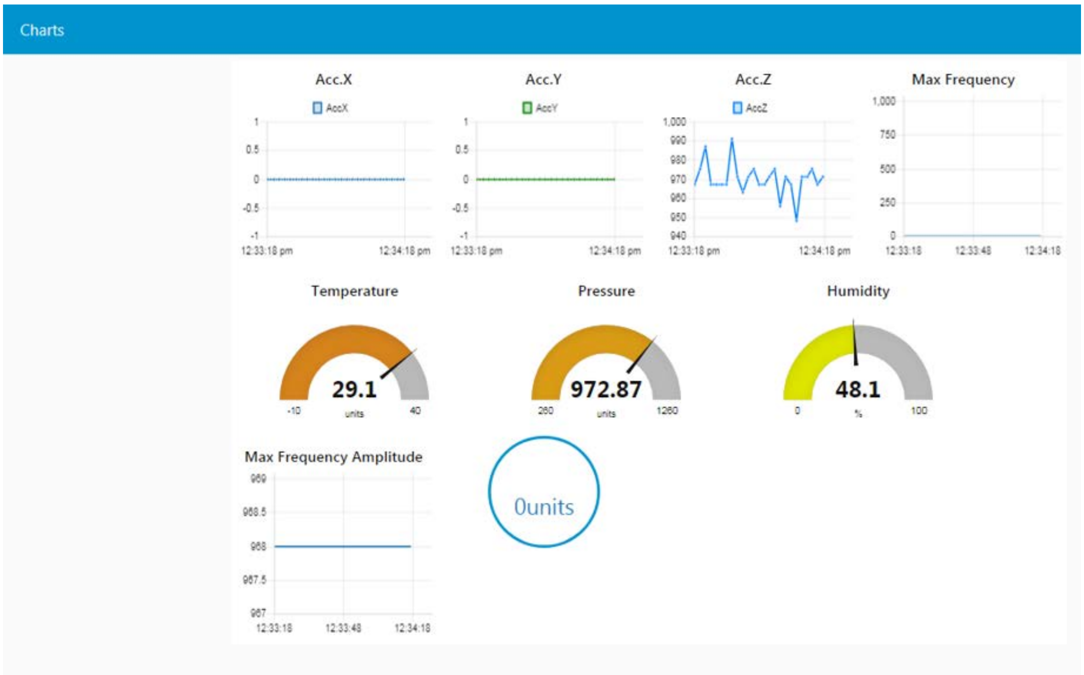
Step 7. To visualize sensor data, click on the [**Deploy**] button on the Node-RED application page top-right, and then click on the [**Launch**] icon in the dashboard tab as shown below.

Figure 34. Deploying Node-RED application



A web-based dashboard appears. If your IoT node device is working properly, sensor data from the STM32 Nucleo are visualized in real-time.

Figure 35. Dashboard for data visualization



3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo platform

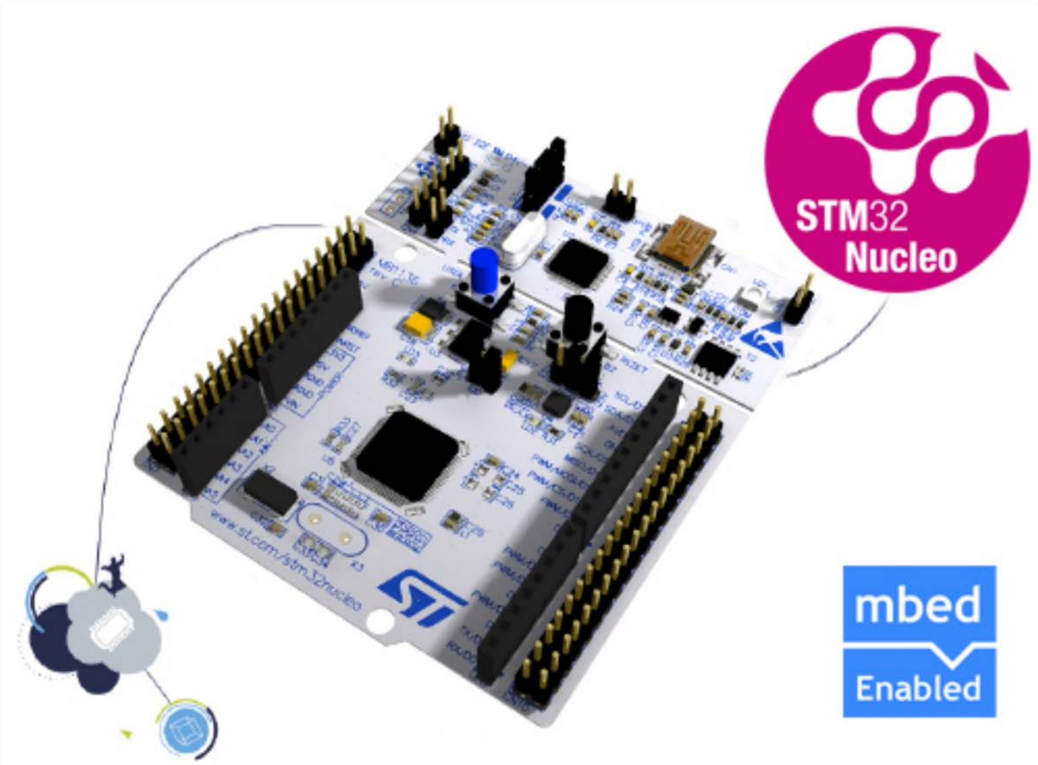
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples.

Figure 36. STM32 Nucleo board



Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

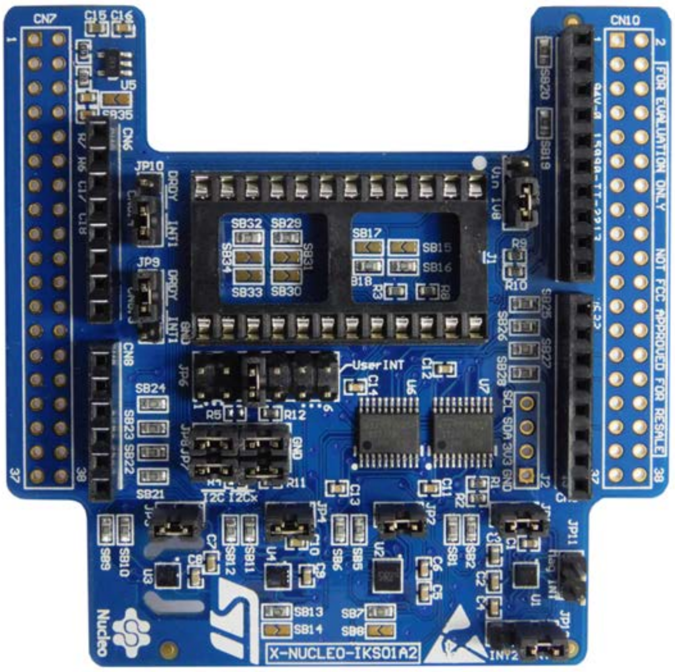
3.1.2 X-NUCLEO-IKS01A2 expansion board

The X-NUCLEO-IKS01A2 is a motion MEMS and environmental sensor expansion board for STM32 Nucleo.

It is compatible with the Arduino UNO R3 connector layout, and is designed around the LSM6DSL 3D accelerometer and 3D gyroscope, the LSM303AGR 3D accelerometer and 3D magnetometer, the HTS221 humidity and temperature sensor and the LPS22HB pressure sensor.

The X-NUCLEO-IKS01A2 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Figure 37. X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board



3.1.3 X-NUCLEO-NFC04A1 expansion board

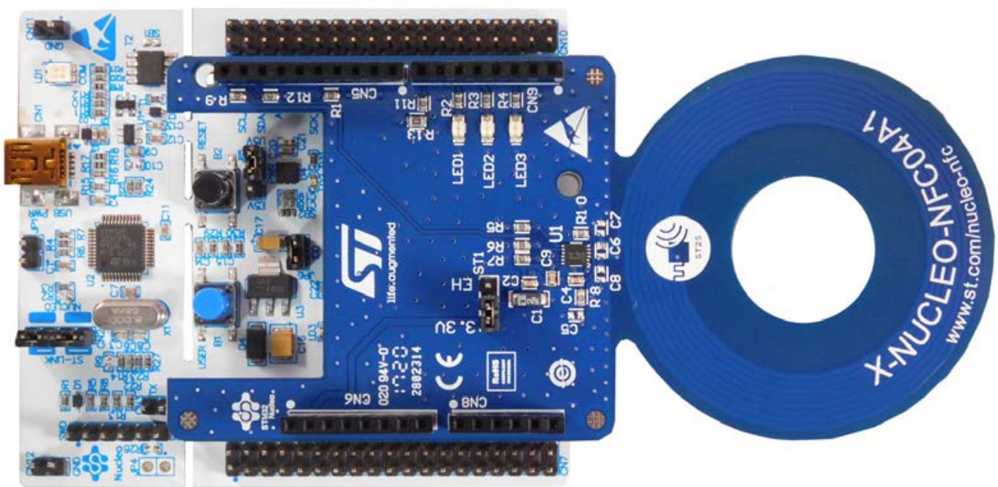
The X-NUCLEO-NFC04A1 dynamic NFC/RFID tag IC expansion board is based on the ST25DV04K NFC Type V/ RFID tag IC with a dual interface 4 Kbits EEPROM that also features an I²C interface. It can be powered by the pin of Arduino connector or directly by the received carrier electromagnetic field.

The X-NUCLEO-NFC04A1 expansion board is compatible with the Arduino™ UNO R3 connector pin assignment and can easily be plugged onto any STM32 Nucleo board. Various expansion boards can also be stacked to evaluate different devices operating together with the dynamic NFC tag.

The board also features an antenna with a 54 mm iso 24.2 diameter, single layer, copper etched on PCB.

Figure 38. **X-NUCLEO-NFC04A1 expansion board**

Figure 39. **X-NUCLEO-NFC04A1 expansion board plugged to an STM32 Nucleo board**



Information about the X-NUCLEO-NFC04A1 expansion board is available on www.st.com at http://www.st.com/x-nucleo

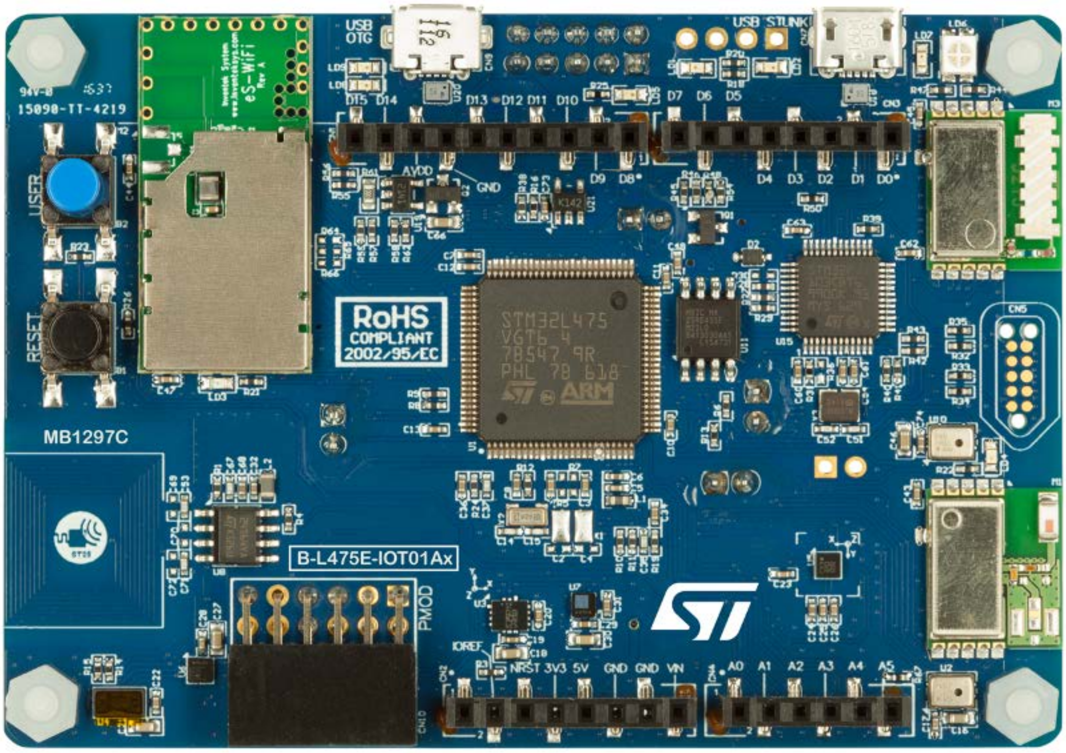
3.1.4 STM32L4 Discovery kit IoT node

The B-L475E-IOT01A Discovery kit for IoT node allows you to develop applications to directly connect to cloud servers.

The Discovery kit enables a wide variety of applications by exploiting low-power communication, multi-way sensing and ARM® Cortex® -M4 core-based STM32L4 series features.

It supports Arduino Uno R3 and PMOD connectivity providing unlimited expansion capabilities with a large choice of dedicated add-on boards.

Figure 40. B-L475E-IOT01A Discovery kit



3.2 Software requirements

To set up a suitable development environment for compiling and running the FP-CLD-WATSON1 package, the following software components are necessary:

- FP-CLD-WATSON1 software available on www.st.com
- Development tool-chain and Compiler, the STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain version 8.20.1 + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - System Workbench for STM32 + ST-LINK
- Serial line monitor (e.g. Tera Term)
- Google Chrome web browser

3.3 Hardware and software setup

3.3.1 Hardware setup

The following hardware components are required:

1. One STM32 Nucleo development platform (order code: NUCLEO-F429ZI)
2. One sensor expansion board (order code: X-NUCLEO-IKS01A2)
3. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
4. Optional: One NFC tag expansion board (order code: X-NUCLEO-NFC04A1)
5. Optional: NFC compatible Android smartphone

3.3.2 Software setup

3.3.2.1 *Development tool-chains and compilers*

Step 1. Select one of the integrated development environments (IDE) supported by the STM32Cube expansion software and listed in Section 3.2 Software requirements, and refer to the system and setup information provided by the selected IDE provider. The project files for all the supported IDEs can be found in the FP-CLD-WATSON1 package (Projects/STM32F429ZI-Nucleo/Applications/Cloud/Bluemix folder).

Step 2. Install a serial line terminal software and a web browser on your PC.

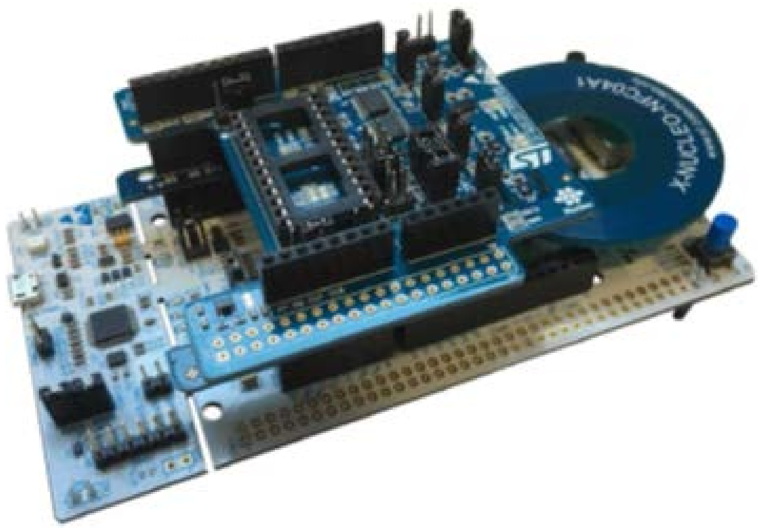
3.3.3 System setup guide

3.3.3.1 *NUCLEO-F429ZI board setup*

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK008 or STSW-LINK009 on www.st.com (according to your version of Microsoft Windows).

The X-NUCLEO-IKS01A2 and X-NUCLEO-NFC04A1 expansion boards can be easily connected to the NUCLEO-F429ZI board through the Arduino UNO R3 extension connector, as shown below.

Figure 41. **NUCLEO-F429ZI connected to X-NUCLEO-NFC04A1 and X-NUCLEO-IKS01A2**



Revision history

Table 2. **Document revision history**

Date	Version	Changes
19-Dec-2016	1	Initial release.
03-Jul-2018	2	Updated all content to reflect software release 3.0.

Contents

1 Acronyms and abbreviations 2

2 FP-CLD-WATSON1 software description 3

2.1 Overview 3

2.2 Features 3

2.3 Architecture 3

2.4 Folder structure 4

2.5 APIs 5

2.6 Sample application description 5

2.6.1 Vibration analysis with LSM303AGR accelerometer 6

2.7 Configure IBM Watson and Wi-Fi access point parameters. 6

2.7.1 Configuration via serial terminal 7

2.7.2 Configuration using NFC 8

2.8 Viewing sensors and vibration analysis data on IBM quickstart web page 12

2.9 Connect to IBM Watson IoT Cloud as a registered device 14

2.10 Connect to Node-RED 19

3 System setup guide 24

3.1 Hardware description 24

3.1.1 STM32 Nucleo platform 24

3.1.2 X-NUCLEO-IKS01A2 expansion board 24

3.1.3 X-NUCLEO-NFC04A1 expansion board 25

3.1.4 STM32L4 Discovery kit IoT node 27

3.2 Software requirements 27

3.3 Hardware and software setup 27

3.3.1 Hardware setup 27

3.3.2 Software setup 28

3.3.3 System setup guide 28

Revision history 29

List of tables

Table 1. List of acronyms . . . 2
Table 2. Document revision history . . . 29

List of figures

Figure 1. FP-CLD-WATSON1 software architecture . . . 4
Figure 2. FP-CLD-WATSON1 package folder structure . . . 5
Figure 3. Configuration of Wi-Fi credentials . . . 7
Figure 4. NUCLEO-F429ZI Ethernet port . . . 7
Figure 5. Enter root CA certificates . . . 8
Figure 8. ST25 NFC app for Android . . . 9
Figure 9. NDEF write menu . . . 10
Figure 10. NDEF selection . . . 11
Figure 11. Write to tag . . . 12
Figure 12. Device streaming sensor data to the IBM Watson Cloud . . . 13
Figure 13. Sensor data shown in the IBM Quickstart web page . . . 13
Figure 14. Automatic opening of IBM quick start web page via NFC . . . 14
Figure 15. Internet of Things Platform Starter Boilerplate: Cloud Foundry App creation . . . 14
Figure 16. Watson IoT Platform service . . . 15
Figure 17. Internet of Things Platform Starter Boilerplate: launching Watson IoT Platform . . . 15
Figure 18. Internet of Things Platform Starter Boilerplate: device type . . . 16
Figure 19. Internet of Things Platform Starter Boilerplate: device name . . . 16
Figure 20. Internet of Things Platform Starter Boilerplate: add device . . . 16
Figure 21. Internet of Things Platform Starter Boilerplate: auto-generated token . . . 17
Figure 22. Internet of Things Platform Starter Boilerplate device ID . . . 17
Figure 23. Internet of Things Platform Starter Boilerplate device credentials . . . 18
Figure 24. Internet of Things Platform Starter Boilerplate registered mode . . . 18
Figure 25. Registration mode device parameter provisioning using ST25 NFC app . . . 19
Figure 26. Node-RED flow editor for IBM Watson IoT Platform . . . 19
Figure 27. Generate API Key and Authentication Token (1 of 3) . . . 20
Figure 28. Generate API Key and Authentication Token (2 of 3) . . . 20
Figure 29. Generate API Key and Authentication Token (3 of 3) . . . 20
Figure 30. Launching Node-RED flow editor . . . 21
Figure 31. Connect device to Node-RED application . . . 21
Figure 32. Configuring IoT node device parameters (1 of 2) . . . 22
Figure 33. Configuring IoT node device parameters (2 of 2) . . . 22
Figure 34. Deploying Node-RED application . . . 23
Figure 35. Dashboard for data visualization . . . 23
Figure 36. STM32 Nucleo board . . . 24
Figure 37. X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board . . . 25
Figure 38. X-NUCLEO-NFC04A1 expansion board . . . 26
Figure 39. X-NUCLEO-NFC04A1 expansion board plugged to an STM32 Nucleo board . . . 26
Figure 40. B-L475E-IOT01A Discovery kit . . . 27
Figure 41. NUCLEO-F429ZI connected to X-NUCLEO-NFC04A1 and X-NUCLEO-IKS01A2 . . . 28