
Getting started with the software package for Wi-Fi, NFC and sensors for connectivity with IBM Bluemix® Cloud in FP-CLD-BLUEMIX1

Introduction

FP-CLD-BLUEMIX1 is an expansion software package for STM32Cube. The software runs on the STM32 microcontroller and includes drivers for the SPWF01SA Serial-to-Wi-Fi module, HTS221, LPS25HB, LSM6DS0, LIS3MDL sensors, and M24SR64-Y dynamic NFC/RFID tag.

It also includes a middleware package implementing the MQTT protocol to facilitate interaction between the STM32 Nucleo boards and Cloud services. The expansion software is built on STM32Cube software technology for portability across different STM32 microcontrollers.

The software includes examples for visualizing sensor data and controlling the device via IBM Bluemix® Cloud services.

Contents

1	Acronyms and abbreviations	5
2	FP-CLD-BLUEMIX1 software description	6
2.1	Overview	6
2.2	Features	6
2.3	Architecture	7
2.4	Folder structure	8
2.5	APIs	8
2.6	Sample application description.....	9
2.6.1	Viewing sensor data on IBM Quickstart	9
2.6.2	Connect IBM Bluemix Cloud as a registered device	11
3	System setup guide.....	13
3.1	Hardware description	13
3.1.1	STM32 Nucleo platform.....	13
3.1.2	X-NUCLEO-IDW01M1 expansion board	13
3.1.3	X-NUCLEO-NFC01A1 expansion board	14
3.1.4	X-NUCLEO-IKS01A1 expansion board.....	15
3.2	Software requirements	16
3.3	Hardware and software setup	16
3.3.1	Hardware setup	16
3.3.2	Software setup.....	17
3.3.3	System setup guide	17
4	Revision history	21

List of tables

Table 1: Acronyms5

Table 2: Document revision history21

List of figures

Figure 1: FP-CLD-BLUEMIX1 software architecture	7
Figure 2: FP-CLD-BLUEMIX1 package folder structure	8
Figure 3: Serial line interface for Wi-Fi configuration (e.g., using Termite)	9
Figure 4: IBM Quickstart URL printed to the serial interface	10
Figure 5: Sensor data view on IBM Quickstart web page.....	10
Figure 6: Automatic opening of web page via NFC	11
Figure 7: Configuration for registering the STM32 device on IBM Bluemix Cloud	12
Figure 8: STM32 Nucleo board.....	13
Figure 9: X-NUCLEO-IDW01M1 Wi-Fi expansion board.....	14
Figure 10: X-NUCLEO-NFC01A1 expansion board	15
Figure 11: X-NUCLEO-IKS01A1 sensor expansion board	16
Figure 12: X-NUCLEO-IDW01M1 Wi-Fi expansion board connected to the STM32 Nucleo via ST morpho connectors	18
Figure 13: X-NUCLEO-NFC01A1 dynamic NFC tag expansion board connected to the STM32 Nucleo on top of the X-NUCLEO-IDW01M1 Wi-Fi expansion board.....	19
Figure 14: X-NUCLEO-IKS01A1 sensors expansion board connected to the STM32 Nucleo on top of NUCLEO-NFC01A1 dynamic NFC tag and X-NUCLEO-IDW01M1 Wi-Fi expansion board.....	20

1 Acronyms and abbreviations

Table 1: Acronyms

Acronym	Description
AP	Access point
BSP	Base support package
FOTA	Firmware update over-the-air
GPIO	General purpose input/output
HAL	Hardware abstraction layer
HTML	Hypertext markup language
HTTP	Hyper text transfer protocol
IDE	Integrated development environment
IoT	Internet of things
I ² C	Inter-integrated circuit
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 transfer 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-BLUEMIX1 software description

2.1 Overview

The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the Wi-Fi, 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, to read/write information from/to NFC/RFID tag and to access and control the Serial-to-Wi-Fi module (SPWF01SA).

The package also includes middleware components implementing the MQ Telemetry Transport (MQTT) application level network protocol for communication with Cloud services. 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.

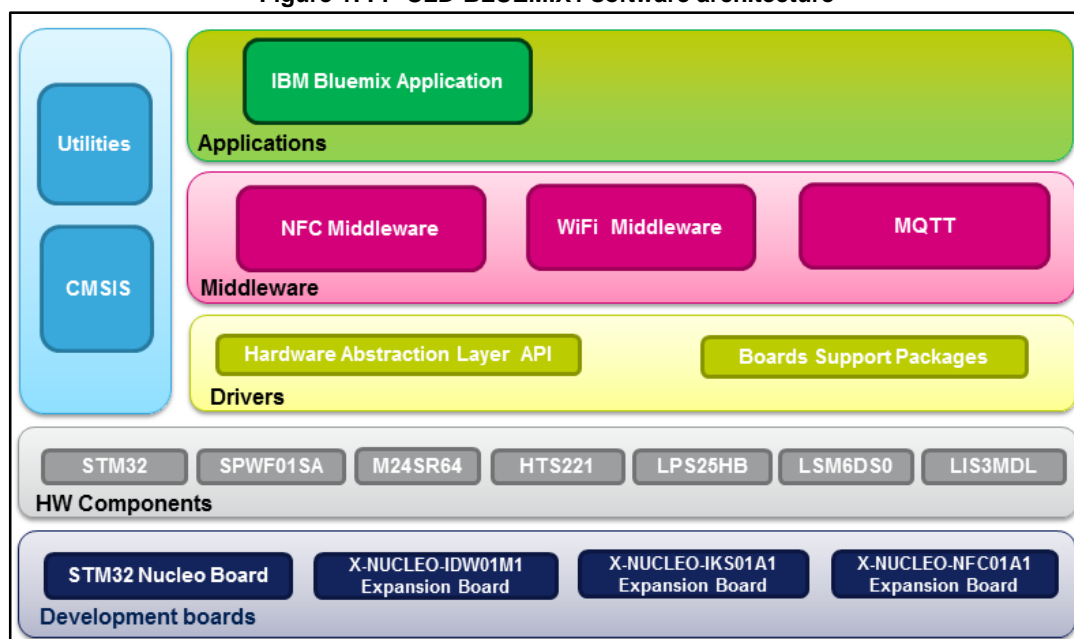
A sample application to experiment connectivity with IBM Cloud is provided on top of the middleware stacks. Developers can use it to prototype end-to-end IoT applications, by registering the STM32 Nucleo microsystem in the IBM Bluemix Cloud service and begin transmitting real time sensor data in a straightforward manner. Thanks to NFC dynamic tag, users can automatically open a trial web page provided by IBM and visualize data generated by all the sensors on the STM32 Nucleo board.

2.2 Features

- Complete middleware to build applications based on Wi-Fi connectivity (SPW01SA) and connect STM32 Nucleo boards with Cloud based services
- Software interface to access the HTS221 temperature and humidity sensor, LPS25HB pressure sensor, and motion sensors LIS3MDL and LSM6DS0, and to write and read the M24SR64-Y RFID/NFC tag
- Integrated MQTT protocol middleware
- Sample implementation available on X-NUCLEO-IKS01A1, X-NUCLEO-IDW01M1, and X-NUCLEO-NFC01A1 boards plugged onto a NUCLEO-F401RE board
- Easy access to IBM Bluemix® Cloud services for sensor data viewing and processing
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

2.3 Architecture

Figure 1: FP-CLD-BLUEMIX1 software architecture



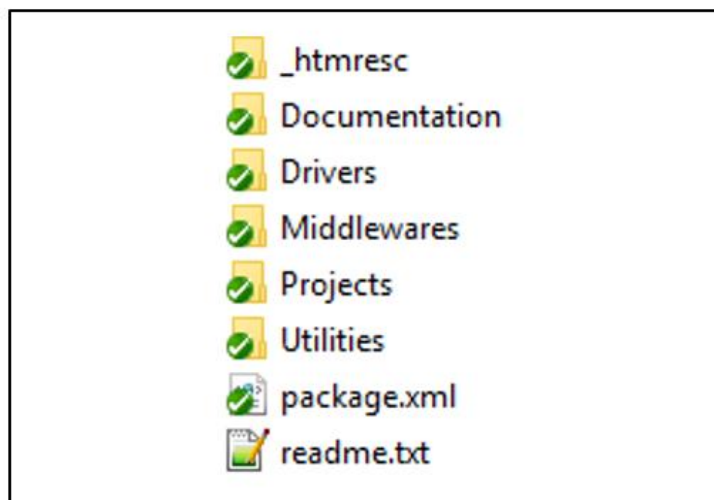
The software layers used by the application to access and use the STM32 microcontroller and the Wi-Fi, sensors and NFC expansion boards are:

- 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 into the STM32Cube package as a middleware library. MQTT is a lightweight messaging protocol with a small code footprint, low power and low bandwidth usage. It is particularly suitable for sensor data telemetry and implementation in embedded systems (more information on the MQTT protocol can be found at www.mqtt.org).

2.4 Folder structure

Figure 2: FP-CLD-BLUEMIX1 package folder structure



The following folders are included in the software package:

- Documentation: contains a compiled HTML file generated from the source code, detailing the software components and APIs.
- Drivers: contains the board-specific HAL drivers for each supported board or hardware platform, including the on-board components and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- Middlewares: contains the middleware interface for Wi-Fi and NFC expansion software, together with MQTT implementation.
- Projects: contains sample applications to connect STM32 Nucleo boards and sensors with IBM Cloud via Wi-Fi.

2.5 APIs

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

The main APIs used by the IBM sample application are:

- `void Config_MQTT_IBM(MQTT_vars * mqtt_ibm_setup , uint8_t * macadd);` to configure the MQTT parameters for connection with IBM Cloud, given the MAC address for the Wi-Fi expansion board.
- `int spwf_socket_create(Network* net, uint8_t * hostname, uint32_t port_number, uint8_t * protocol);` to open a socket given a hostname, port number and protocol type (TCP or TLS). It returns success or failure and socket id within the network structure.
- `void MQTTClient(Client* c, Network* network, unsigned intcommand_timeout_ms, unsigned char* buf, size_tbuf_size, unsigned char* readbuf, size_treadbuf_size);` to configure the Client structure given the input parameters.
- `int MQTTConnect(Client* c, MQTTPacket_connectData* options);` to connect with the IBM MQTT broker given the parameters in Client and MQTTPacket_connect data structure. It returns success or failure.

- `int MQTTSubscribe(Client* c, const char* topicFilter, enumQoSqos, messageHandlermessageHandler);` to subscribe to the MQTT topic defined in data structure Client. It returns success or failure.
- `int MQTTPublish (Client* c, const char* topicName, MQTTMessage* message);` to publish a message to the MQTT topic defined in topicName. It returns success or failure.
- `int MQTTYield (Client* c, inttimeout_ms);` to check whether a message has been received by the client in a given time slot. It returns success or failure.

2.6 Sample application description

A sample application using the X-NUCLEO-IKS01A1, X-NUCLEO-IDW01M1 and X-NUCLEO-NFC01A1 expansion boards with the NUCLEO-F401RE board is provided 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 these to the IBM IoT Cloud via Wi-Fi. The URL of the web page where the sensor data can be visualized is written in the NFC tag.

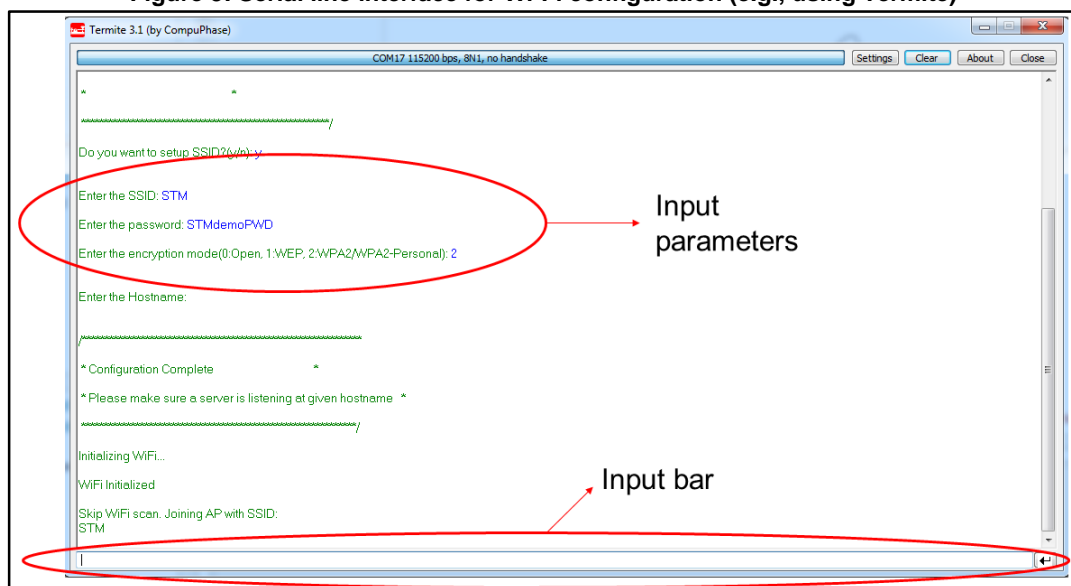
The application is configured by default to run in Quickstart mode for data visualization only. However, it is easily modified to register and control the device in IBM Cloud (requires an account on IBM Bluemix Cloud).

2.6.1 Viewing sensor data on IBM Quickstart

The sample application for FP-CLD-BLUEMIX1 package contains a default configuration to connect the STM32 Nucleo microsystem to IBM Cloud to view sensor data on an IBM webpage (<https://quickstart.internetofthings.ibmcloud.com>), according to the MAC address of the device. By default, the unique URL page where the data is published is also written in the NFC tag.

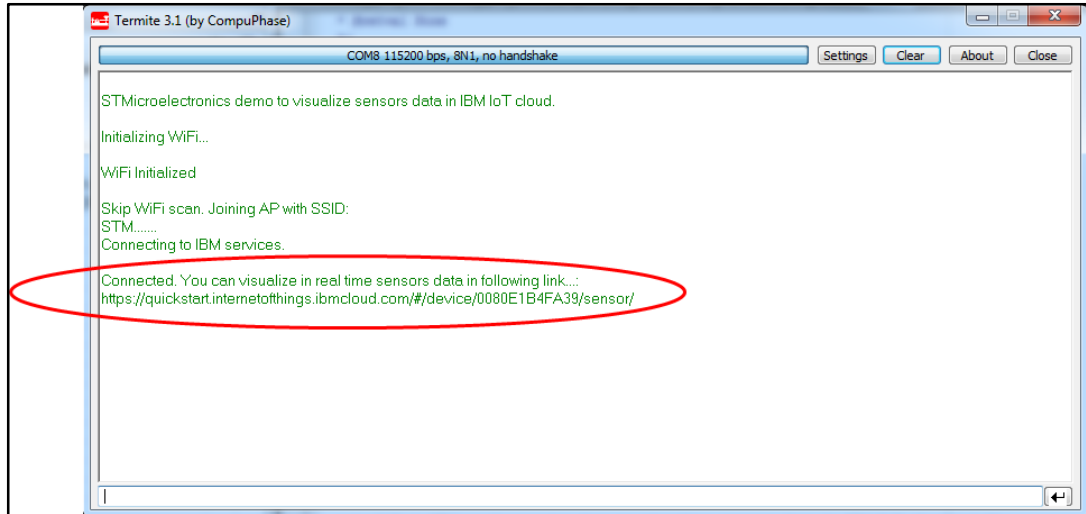
Once the setup is configured and the application is launched as described in [Section 3.3.3: "System setup guide"](#), you need a serial line monitor to visualize messages from the STM32 Nucleo microsystem and configure the SSID and Password to connect the board with a Wi-Fi network, as per the figure below.

Figure 3: Serial line interface for Wi-Fi configuration (e.g., using Termite)



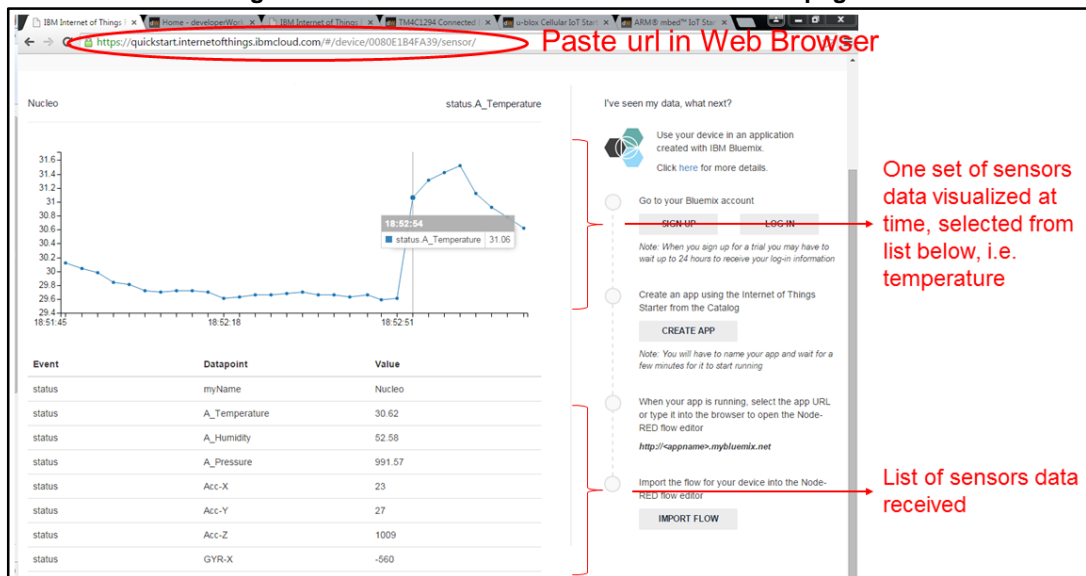
Once connected with a Wi-Fi access point, the application prints the MAC address of the Wi-Fi expansion board and the IBM Quickstart URL to the screen.

Figure 4: IBM Quickstart URL printed to the serial interface



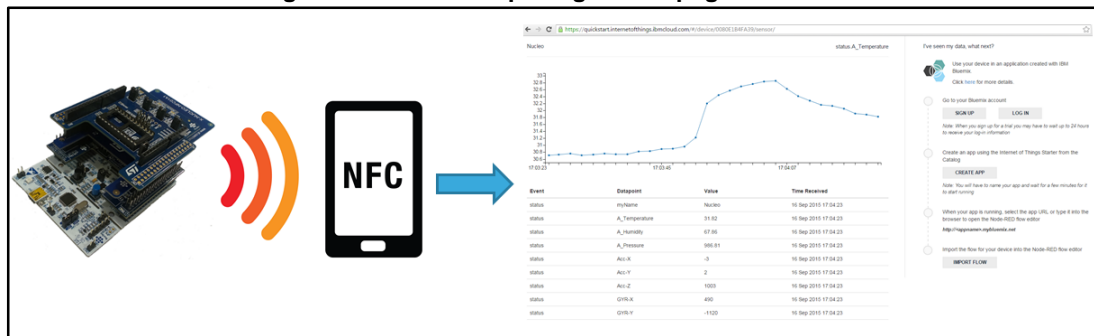
This URL can be copied and pasted in a web browser to view the real time sensor data, as shown below.

Figure 5: Sensor data view on IBM Quickstart web page



If an NFC-enabled mobile device is available, the Quickstart web page can be even more easily opened by nearing the device to the NFC tag, as shown below.

Figure 6: Automatic opening of web page via NFC



2.6.2 Connect IBM Bluemix Cloud as a registered device

In order to register the STM32 Nucleo microsystem on IBM Bluemix Cloud, you must first create an account on Bluemix by following these instructions:

<https://console.ng.bluemix.net/>

Once you have signed onto IBM Bluemix Cloud and selected the Internet of Things service, you can register your device. During the device registration procedure, several device properties are provided:

- Organization ID (i.e., "1w8a05").
- Device type (i.e., "stm32_nucleo").
- Authentication Method (only "use-token-auth" supported).
- Authentication token (i.e., "uUURNRbeQQaX+Fvi&8").

These properties must then be inserted in the `Config_MQTT_IBM` function in the 'IBM_Bluemix_Config.c' file of the 'Projects/Multi/Applications/MQTT_IBM/Src' folder'. In the same function, set the `ibm_mode` variable to `REGISTERED` as shown in the figure below.

Figure 7: Configuration for registering the STM32 device on IBM Bluemix Cloud

```

void Config_MQTT_IBM ( MQTT_vars *mqtt_ibm_setup, uint8_t *macadd )
{
    /* Default Configuration for QUICKSTART. REGISTERED mode requires account on Bluemix */
    mqtt_ibm_setup->ibm_mode = REGISTERED;
    /* Quickstart visualization */
    if ( mqtt_ibm_setup->ibm_mode == QUICKSTART )
    {
        strcpy((char*)mqtt_ibm_setup->pub_topic, "iot-2/evt/status/fmt/json");
        strcpy((char*)mqtt_ibm_setup->sub_topic, "");
        strcpy((char*)mqtt_ibm_setup->clientid, "d:quickstart:nucleo:");
        strcat((char*)mqtt_ibm_setup->clientid, (char *)macadd);
        mqtt_ibm_setup->qos = QOS0;
        strcpy((char*)mqtt_ibm_setup->username, "");
        strcpy((char*)mqtt_ibm_setup->password, "");
        strcpy((char*)mqtt_ibm_setup->hostname, "quickstart.messaging.internetofthings.ibmcloud.com");
        strcpy((char*)mqtt_ibm_setup->device_type, "");
        strcpy((char*)mqtt_ibm_setup->org_id, "");
        mqtt_ibm_setup->port = 8883; //TLS
        mqtt_ibm_setup->protocol = 's'; // TLS no certificates
    }
    else
    {
        /* REGISTERED DEVICE */
        /* Need to be customized */
        strcpy((char*)mqtt_ibm_setup->pub_topic, "iot-2/evt/status/fmt/json");
        strcpy((char*)mqtt_ibm_setup->sub_topic, "iot-2/cmd/+ /fmt/string");
        mqtt_ibm_setup->qos = QOS0;
        strcpy((char*)mqtt_ibm_setup->username, "use-token-auth"); // Customize
        strcpy((char*)mqtt_ibm_setup->password, "uUURNRbeQQaX+Fvis8");
        strcpy((char*)mqtt_ibm_setup->hostname, "lw8a05.messaging.internetofthings.ibmcloud.com");
        strcpy((char*)mqtt_ibm_setup->device_type, "stm32_nucleo");
        strcpy((char*)mqtt_ibm_setup->org_id, "lw8a05");
        strcpy((char *)mqtt_ibm_setup->clientid, "d:");
        strcat((char *)mqtt_ibm_setup->clientid, (char *)mqtt_ibm_setup->org_id);
        strcat((char *)mqtt_ibm_setup->clientid, ":");
        strcat((char *)mqtt_ibm_setup->clientid, (char *)mqtt_ibm_setup->device_type);
        strcat((char *)mqtt_ibm_setup->clientid, ":");
        strcat((char*)mqtt_ibm_setup->clientid, (char *)macadd);
        mqtt_ibm_setup->port = 8883; //TLS
        mqtt_ibm_setup->protocol = 's'; // TLS no certificates
    }
    return;
}

```

Once registered and connected to IBM Cloud, the STM32 Nucleo microsystem can send and receive information to and from IBM Cloud applications and IoT services. More information about how to develop Cloud applications and services are available at:

<https://www.ng.bluemix.net/docs/starters/IoT/iot500.html>

3 System setup guide

3.1 Hardware description

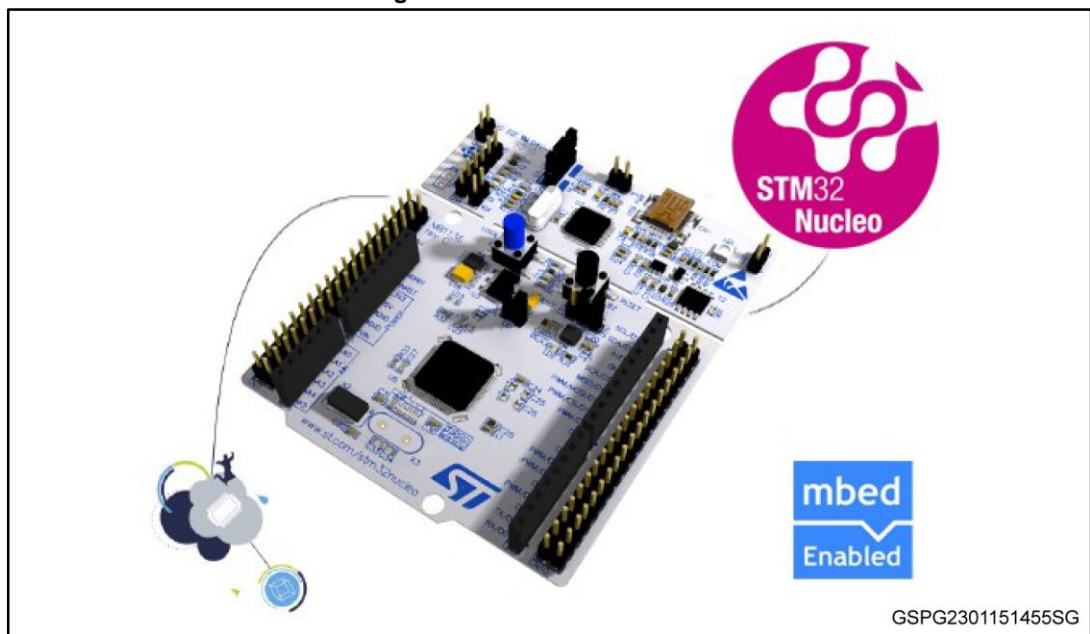
This section describes the hardware components needed to connect STM32 based platform to IBM Cloud.

3.1.1 STM32 Nucleo platform

The STM32 Nucleo boards provide an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller lines. The Arduino™ connectivity support and ST morpho headers 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 any separate probe 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.

Information regarding STM32 Nucleo boards is available on www.st.com at <http://www.st.com/stm32nucleo>

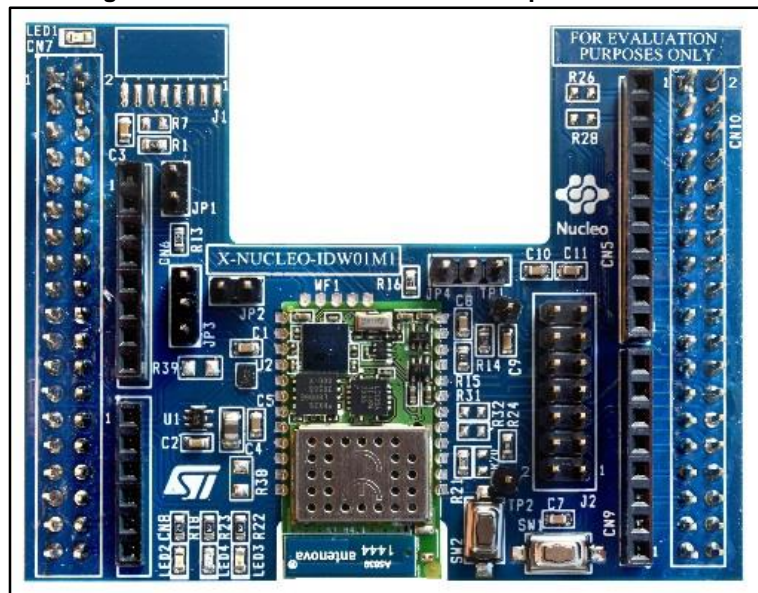
Figure 8: STM32 Nucleo board



3.1.2 X-NUCLEO-IDW01M1 expansion board

Figure 9: "X-NUCLEO-IDW01M1 Wi-Fi expansion board" shows the Wi-Fi board based on the SPWF01SA module, which expands the STM32 Nucleo range. The CE, IC and FCC-certified SPWF01SA module has an embedded STM32 MCU, a low-power Wi-Fi b/g/n SoC with integrated power amplifier and power management, and an SMD antenna. The SPWF01SA module is also equipped with 1 MByte of external FLASH for firmware update over-the-air (FOTA). The module can simultaneously behave as a socket server and socket client. The X-NUCLEO-IDW01M1 interfaces with the MCU on the STM32 Nucleo board via the UART serial port. The user can easily access the stack functions using the AT command. X-NUCLEO-IDW01M1 is compatible with both the ST morpho and Arduino UNO R3 connector layout.

Figure 9: X-NUCLEO-IDW01M1 Wi-Fi expansion board



More information about the X-NUCLEO-IDW01M1 Wi-Fi expansion board is available on www.st.com.

3.1.3 X-NUCLEO-NFC01A1 expansion board

Figure 10: "X-NUCLEO-NFC01A1 expansion board" shows the Dynamic NFC tag expansion for STM32 Nucleo boards. It is compatible with the Arduino UNO R3 connector layout and is designed around the M24SR64-Y dynamic NFC/RFID tag IC with dual interface. It embeds a 64 Kbit EEPROM and can be operated from an I²C interface or a 13.56 MHz RFID reader or an NFC phone. The board is powered through the Arduino UNO R3 connectors. It also has three general purpose LEDs.

Figure 10: X-NUCLEO-NFC01A1 expansion board



More information about the X-NUCLEO-NFC01A1 expansion board is available on www.st.com.

3.1.4 X-NUCLEO-IKS01A1 expansion board

Figure 11: "X-NUCLEO-IKS01A1 sensor expansion board" shows the sensor expansion board for STM32 Nucleo. It is compatible with the Arduino UNO R3 connector layout and is designed around the STMicroelectronics temperature and humidity (HTS221), pressure (LPS25HB) and motion (LIS3MDL and LSM6DS0) sensing devices. The X-NUCLEO-IKS01A1 interfaces with the STM32 MCU via an I²C pin, and the user can change the default I²C port and device IRQ by changing one resistor on the evaluation board.

You can connect the LSM6DS3 DIL24 expansion and use it instead of the one for LSM6DS0.

Figure 11: X-NUCLEO-IKS01A1 sensor expansion board



More information about the X-NUCLEO-IKS01A1 expansion board is available on [st.com](http://www.st.com).

3.2 Software requirements

To set up a suitable development environment for compiling and running the FP-CLD-BLUEMIX1 package, the following software components are necessary:

- FP-CLD-BLUEMIX1 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 + ST-LINK
 - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
 - System Workbench for STM32 + ST-LINK
- Serial line monitor (e.g., Termit, http://www.compuphase.com/software_termite.htm)
- Chrome web browser (<http://www.google.com/chrome/>)

3.3 Hardware and software setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

3.3.1 Hardware setup

The following hardware components are required:

1. One STM32 Nucleo Development platform (order code: NUCLEO-F401RE)
2. One Wi-Fi expansion board (order code: X-NUCLEO-IDW01M1)
3. One NFC tag expansion board (order code: X-NUCLEO-NFC01A1)
4. One sensors expansion board (order code: X-NUCLEO-IKS01A1)
5. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

3.3.2 Software setup

This section lists the minimum requirements for the developer to setup the FP-CLD-BLUEMIX1 SDK, run the sample testing scenario and customize applications.

3.3.2.1 Development Tool-chains and Compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software and listed above and follow the system and setup information provided by the selected IDE provider. Project files for all three supported IDEs can be found in the `Projects/Multi/Applications/MQTT_IBM` folder of the FP-CLD-BLUEMIX1 package.

3.3.3 System setup guide

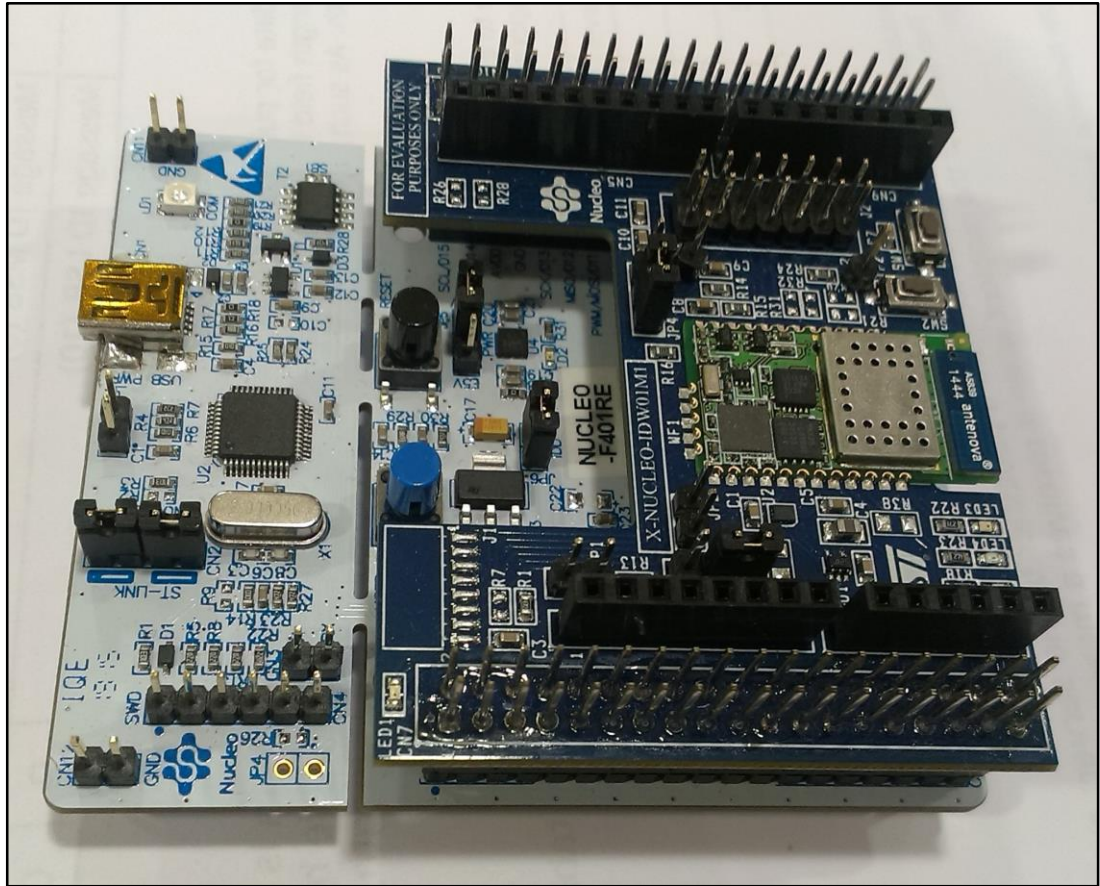
This section describes how to set up different hardware parts before writing and executing an application on the STM32 Nucleo microsystem.

3.3.3.1 STM32 Nucleo and Wi-Fi expansion boards 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).

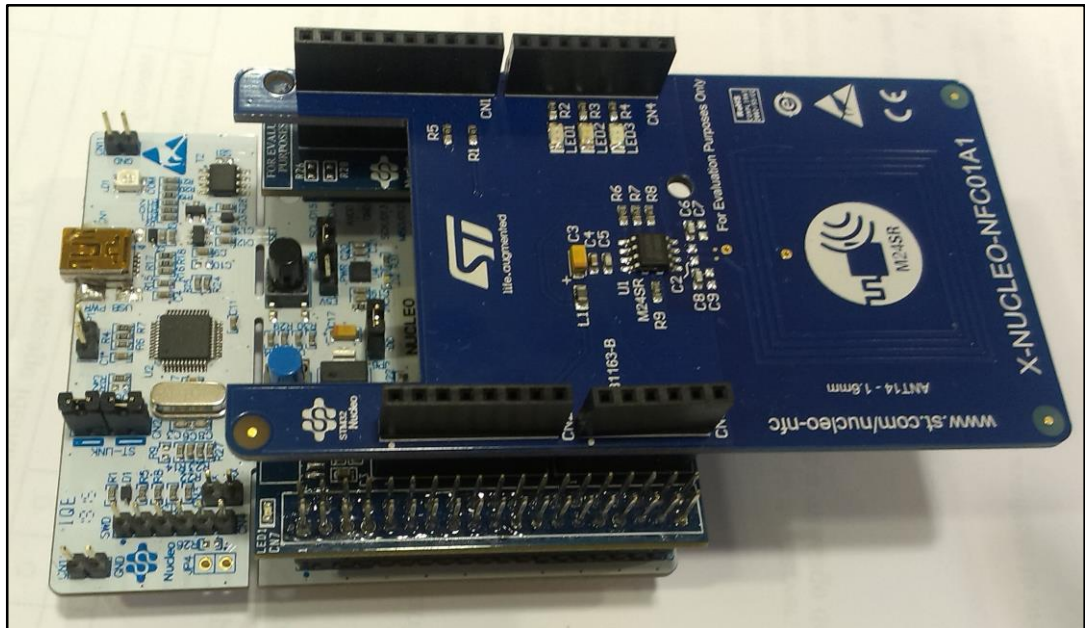
The Wi-Fi expansion board X-NUCLEO-IDW01M1 can be connected to the STM32 Nucleo through the ST morpho extension connector, as shown below.

Figure 12: X-NUCLEO-IDW01M1 Wi-Fi expansion board connected to the STM32 Nucleo via ST morpho connectors



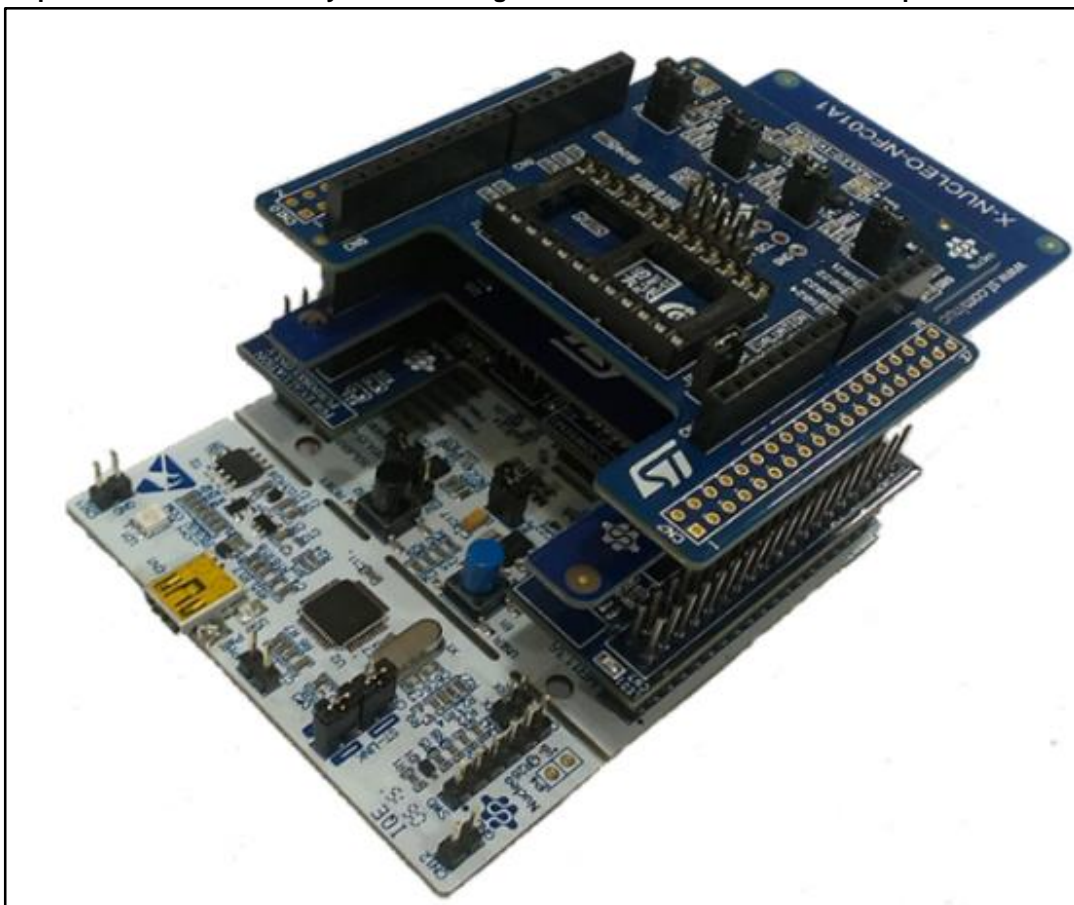
The Dynamic NFC tag board X-NUCLEO-NFC01A1 can be connected to X-NUCLEO-IDW01M1 expansion board through the Arduino UNO R3 extension connector, as shown below.

Figure 13: X-NUCLEO-NFC01A1 dynamic NFC tag expansion board connected to the STM32 Nucleo on top of the X-NUCLEO-IDW01M1 Wi-Fi expansion board



Finally, the X-NUCLEO-IKS01A1 sensors board is easily connected to the X-NUCLEO-NFC01A1 expansion board through the Arduino UNO R3 extension connector, as shown below.

Figure 14: X-NUCLEO-IKS01A1 sensors expansion board connected to the STM32 Nucleo on top of NUCLEO-NFC01A1 dynamic NFC tag and X-NUCLEO-IDW01M1 Wi-Fi expansion board



4 Revision history

Table 2: Document revision history

Date	Version	Changes
21-Jan-2016	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2016 STMicroelectronics – All rights reserved