

Getting started with the BlueMS 3.x application

Introduction

The ST BlueMS mobile app has been developed to support a set of STM32Cube expansion software packages (function packs) based on the BlueST BLE protocol.

The application is available for both Android and iOS platforms and can be found here: [Android](#), [iOS](#).

The application source code is released under a BSD license and is available for both the platforms on GitHub: [BlueMS-Android-src](#), [BlueMS-iOS-src](#).

It supports the following expansion software packages: [BlueMicrosystem1](#) version 2.0 and above, [BlueMicrosystem2](#) version 1.0 and above, [BlueMicrosystem3](#) version 1.0 and above, [FP-SNS-ALLMEMS1](#) version 1.0 and above, [FP-SNS-MOTENV1](#) version 1.0 and above and [FP-SNS-FLIGHT1](#) version 1.0 and above, [STSW-WESU1](#) version 1.0 and above.

The BlueST protocol allows to easily receive the data over Bluetooth Low Energy and can be extended to support new data types. The SDK source code is released under a BSD license and available on GitHub: [BlueMS-SDK-Aar](#), [BlueMS-SDK-Ipa](#).

Contents

1	References, acronyms and abbreviations	6
2	Setup	7
3	Mobile application	8
3.1	Demos.....	8
3.1.1	Environmental.....	10
3.1.2	MEMS Sensor Fusion.....	11
3.1.3	Plot data	14
3.1.4	Activity recognition.....	15
3.1.5	Carry position	16
3.1.6	Gesture recognition	17
3.1.7	Pedometer	18
3.1.8	Acceleration events	19
3.1.9	LED switch.....	22
3.1.10	BlueVoice	22
3.1.11	Compass	27
3.1.12	Motion intensity.....	28
3.1.13	Rssi and battery.....	29
3.1.14	Cloud	29
3.2	Secure pairing	31
3.2.1	NFC	32
3.3	Debug console	33
3.4	Log settings.....	33
3.5	License Manager.....	35
3.5.1	Requesting a license	36
3.5.2	Uploading the license	37
3.6	Firmware upgrade	38
4	Google speech ASR Key generation.....	42
5	IBM Watson IoT	45
5.1	Register a device	45
5.2	Display data	46
6	Revision history	47

List of tables

Table 1: References.....6

Table 2: Acronyms6

Table 3: Document revision history47

List of figures

Figure 1: Main application view	8
Figure 2: Android application scanning results	8
Figure 3: The application asks to send the acquired log by e-mail	9
Figure 4: The serial console is active and shows the data sent by the device	9
Figure 5: Example demo menu	10
Figure 6: Environmental page	11
Figure 7: MEMS Sensor Fusion page	12
Figure 8: MEMS Sensor fusion calibration dialog	13
Figure 9: MEMS Sensor fusion reset board position	13
Figure 10: Plot Data page	14
Figure 11: Selecting the data to plot	15
Figure 12: Selecting the time scale	15
Figure 13: Activity recognition demo page	16
Figure 14: Carry position demo page	17
Figure 15: Mems gesture demo page	18
Figure 16: Pedometer demo page	19
Figure 17: Orientation event	20
Figure 18: Free fall event	20
Figure 19: Single Tap event	20
Figure 20: Double Tap event	20
Figure 21: Wake Up event	21
Figure 22: Pedometer event	21
Figure 23: Tilt event	21
Figure 24: Switch demo page	22
Figure 25: BlueVoice Demo page	23
Figure 26: List of supported languages	24
Figure 27: Add the service key request	24
Figure 28: Insert key dialog box	24
Figure 29: Google service enabled	25
Figure 30: Google service recording a message	25
Figure 31: iFlytek service enabled	26
Figure 32: iFlytek demo during the streaming	26
Figure 33: Compass demo page	27
Figure 34: Motion intensity demo page	28
Figure 35: Board status demo page	29
Figure 36: Data needed for the Quickstart service	30
Figure 37: List of available features after the connection	30
Figure 38: IBM Watson Quickstart webpage	31
Figure 39: PIN dialog prompt	32
Figure 40: Debug serial console	33
Figure 41: Settings screen	34
Figure 42: Path to save the log data	34
Figure 43: Dialog to select the log format	34
Figure 44: Main License Manager view	35
Figure 45: License Manager menu	35
Figure 46: Example of license to agree with	36
Figure 47: Fields to compile to request the license	36
Figure 48: Screen where the license code must be pasted	37
Figure 49: Text to copy and paste in BlueMS	37
Figure 50: License loaded confirmation message	38
Figure 51: Firmware upgrade main view	39
Figure 52: Formatting the flash memory	40
Figure 53: Uploading the new firmware	40

Figure 54: Uploading complete with success	41
Figure 55: Error during the uploading	41
Figure 56: Google Chromium-dev: search group	42
Figure 57: Google Chromium-dev: join group to post.....	42
Figure 58: Google Chromium-dev: join the group	42
Figure 59: Google Chromium-dev: create project	43
Figure 60: Google Developers Console: new project	43
Figure 61: Google Developers Console: ASRProject.....	43
Figure 62: Google Developers Console: select API	43
Figure 63: Google API Manager: enable API	44
Figure 64: Google API Manager: create API key	44
Figure 65: Google API Manager: Android API key	44
Figure 66: Watson IoT platform: Add Device.....	45
Figure 67: Watson IoT platform: Choose Device Type.....	45
Figure 68: Watson IoT platform: Add New Card.....	46
Figure 69: Watson IoT platform: add a new dataset.....	46

1 References, acronyms and abbreviations

Table 1: References

Order code	Description
BlueST-SDK-Ipa	iOS version of BlueST SDK library that permits easy access to the data exported by a Bluetooth Low Energy (BLE) device that implements the BlueST protocol.
BlueST-SDK-Aar	Android version of BlueST SDK library that permits easy access to the data exported by a Bluetooth Low Energy (BLE) device that implements the BlueST protocol.
BlueMS-iOS	iOS demo application compatible with BlueST SDK protocol compatible STM32Cube expansion software.
BlueMS-Android	Android demo application compatible with BlueST SDK protocol compatible STM32Cube expansion software.
BLUEMICROSYSTEM1	Bluetooth low energy and sensors expansion software for STM32Cube
BLUEMICROSYSTEM2	Bluetooth low energy and sensors expansion software for STM32Cube
BLUEMICROSYSTEM3	Bluetooth low energy, sensors and NFC tag software expansion for STM32Cube
FP-SNS-MOTENV1	Bluetooth Low Energy and sensors software expansion for STM32Cube. STM32 ODE version of the BLUEMICROSYSTEM1
FP-SNS-ALLMEMS1	Bluetooth Low Energy and sensors software expansion for STM32Cube. STM32 ODE version of the BLUEMICROSYSTEM2
FP-SNS-FLIGHT1	Bluetooth Low Energy and sensors software expansion for STM32Cube. STM32 ODE version of the BLUEMICROSYSTEM3

Table 2: Acronyms

Acronym	Description
ASR	Automatic speech recognition
BLE	Bluetooth low energy
CSV	Coma separated values
GPU	Graphics processing unit
MCU	Microcontroller unit
MEMS	MicroElectroMechanical systems
RSSI	Received signal strength indication
NFC	Near field communication
SDK	Software development kit

2 Setup

The user needs an STM32 Nucleo board or/and the relevant expansion boards to set up the chosen expansion software pack.^a

It is also possible to connect the STEVAL-WESU1 or the STEVAL-STLKT01V1 evaluation boards.

iOS 9 or above compatible application can be found at: [BlueMS-iOS](#).

Android 4.4 or above devices providing Bluetooth Low Energy functionality compatible application can be found at: [BlueMS-Android](#).

^a Refer to the proper expansion software page at <http://www.st.com/STM32ODE-FP> for further information.

3 Mobile application

On the initial application screen, you can access customary application information and start a scan sequence for compatible BLE nodes in the proximity.

Figure 1: Main application view

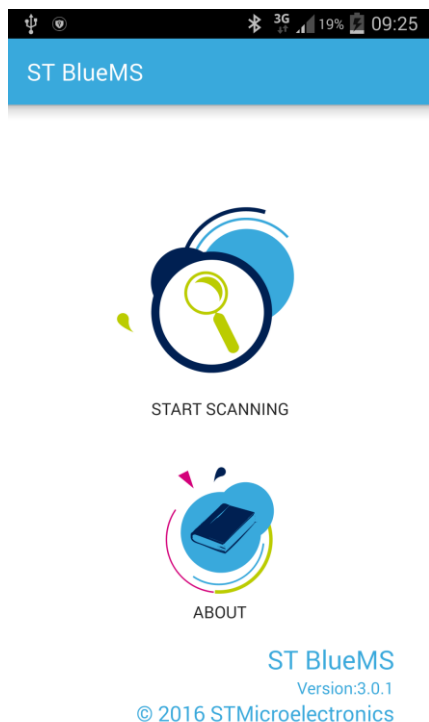
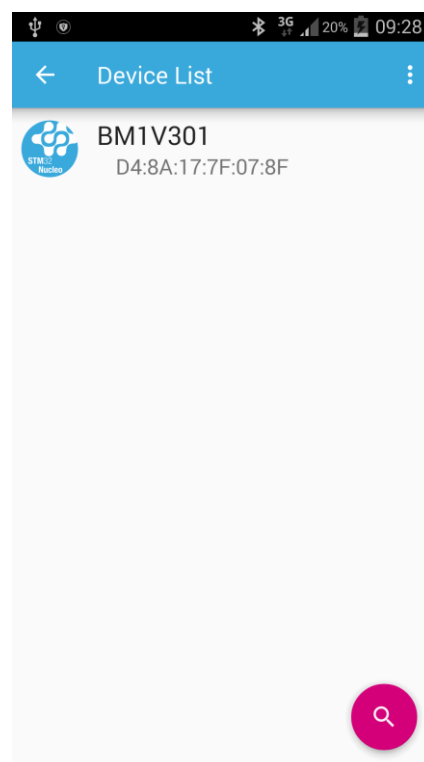


Figure 2: Android application scanning results



On Android 6.0 and above, the app needs permission to read the current device location. This information is not used by the app but is needed to the system to initiate Bluetooth discovery.

When the scan finishes, select a device to connect with and the application switches to the demo view and starts the connection procedure.

3.1 Demos

All the demos share certain features such as the ability to log data and show the output of the serial console.

- 1 Select the START LOGGING button top menu bar in the application menu.
The data is logged by default in one CSV file per data stream. When acquisition stops, the application prompts the user to confirm whether or not to forward the data by e-mail.

- 2 Show the serial console in the bottom of the page (Android only) by clicking Show Serial Console in the overflow menu.
To hide the console, click Hide Serial Console.

Figure 3: The application asks to send the acquired log by e-mail

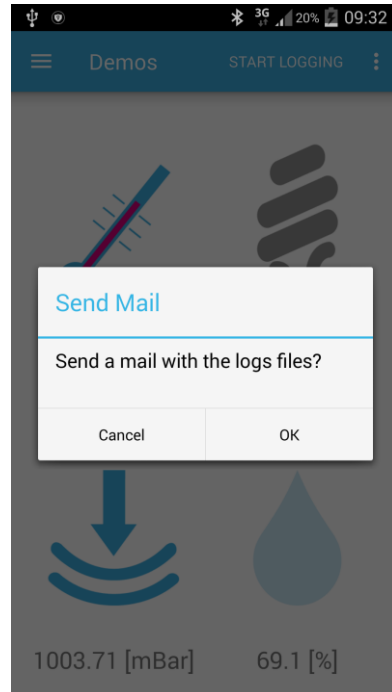
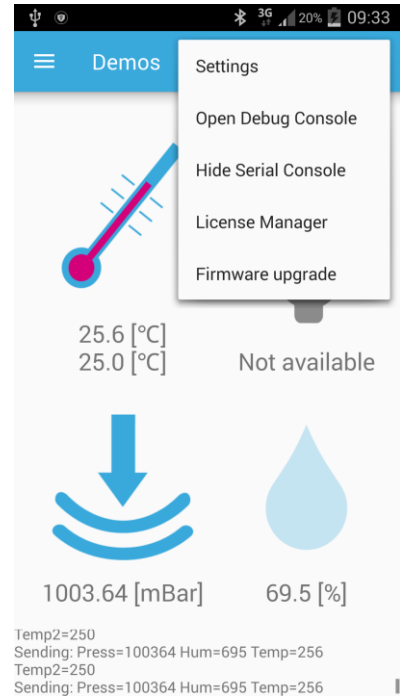


Figure 4: The serial console is active and shows the data sent by the device

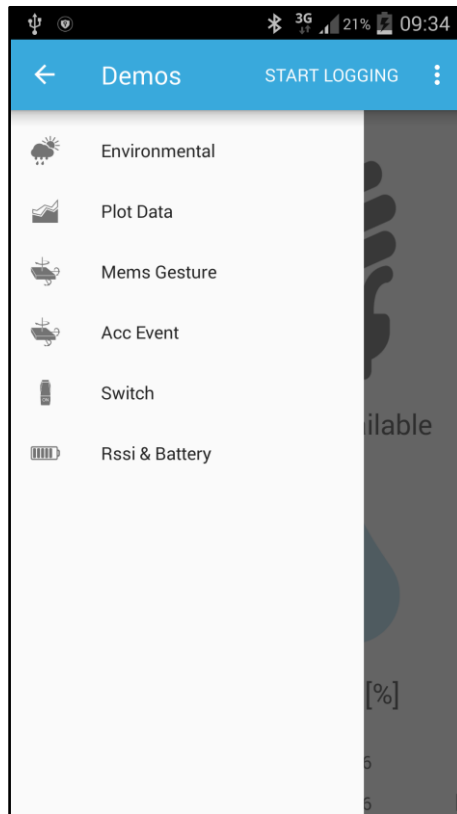


- 3 Swipe from the left border of the screen or tap on the top left icon to reveal all the available demos.
You can also sweep through the available views with a simple left/right screen swipe gesture.



Your list may differ from [Figure 5: "Example demo menu"](#) as this menu only shows the demo supported by your current device/firmware.

Figure 5: Example demo menu



- 4 Clicking on a menu item will show the specific demo view.

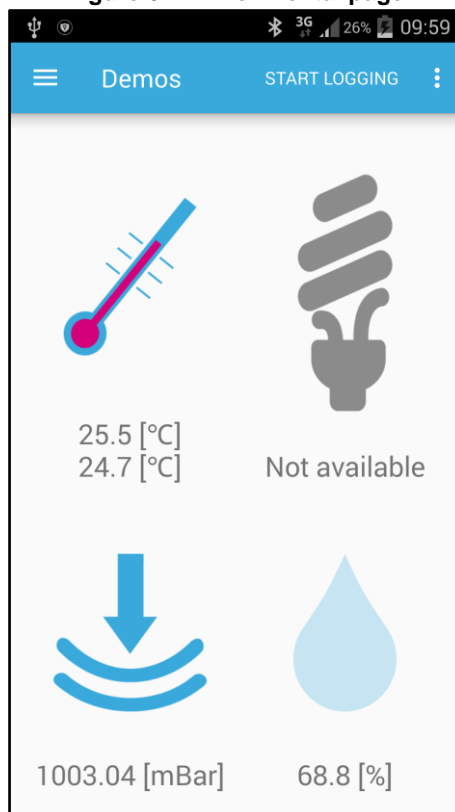
3.1.1 Environmental

This page shows the data coming from the X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 (temperature, humidity and pressure) and X-NUCLEO-6180XA1 (luminosity) expansion boards.

If one of these boards is not present, the related data and icon(s) are disabled.

- 1 Click on an image to force the data reading.
Multiple data streams of the same type are supported: in the figure below, the first temperature reading is from the humidity/temperature sensor (HTS221), while the second is from the pressure/temperature sensor (LPS25HB) .

Figure 6: Environmental page



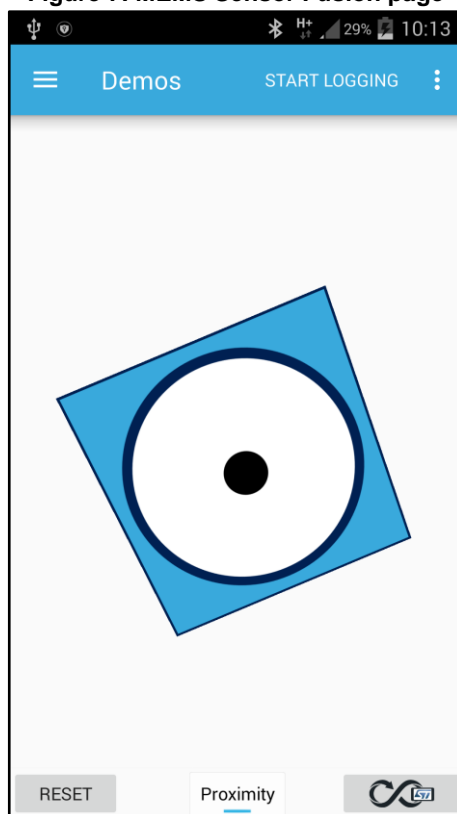
3.1.2 MEMS Sensor Fusion

To enable this demo a compatible firmware is needed and the user must obtain and load a valid MotionFX license.^a

This page shows a 3D cube that reflects board motion thanks to the output of the sensor fusion algorithm on the STM32 Nucleo board that calculates rotation using the magnetometer, accelerometer and gyroscope data from the X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 expansion board.

^a See [Section 3.5: "License Manager"](#).

Figure 7: MEMS Sensor Fusion page

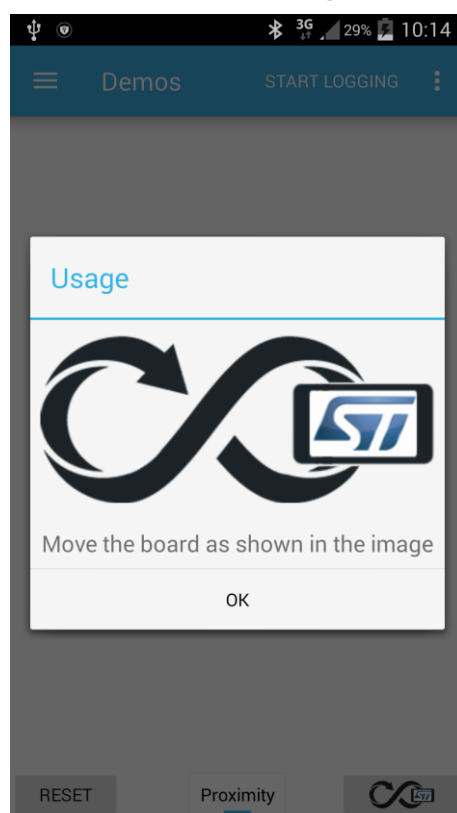
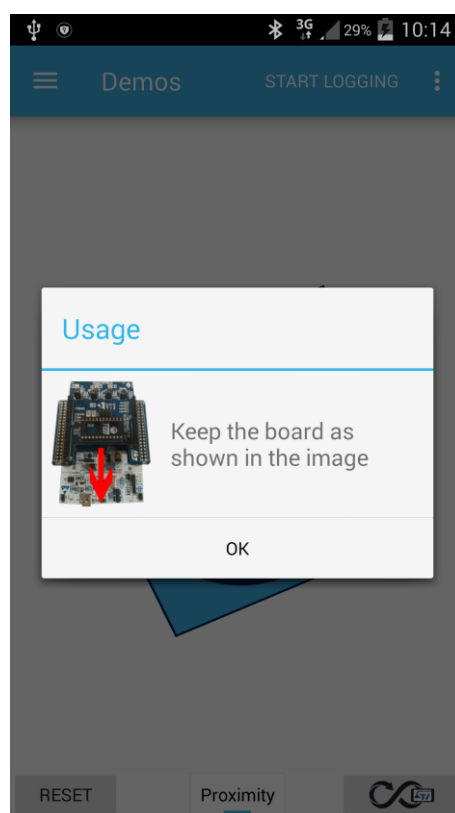


If the X-NUCLEO-6180XA1 expansion board is also present, the cube zoom factor will vary according to the distance read from the proximity sensor.

You can disable and enable the proximity function by pressing the Proximity button.

- 1 Run the calibration process to improve the precision of the sensor fusion algorithm; pushing the bottom right button deletes existing calibration data and starts a new calibration process.
As the dialog suggests, you must rotate the board in a figure 8 pattern.
The board is calibrated when the bottom right button becomes green.
- 2 Align the starting position to ensure coherent movement between the board and the cube: press the reset button and keep the board as shown in the image.

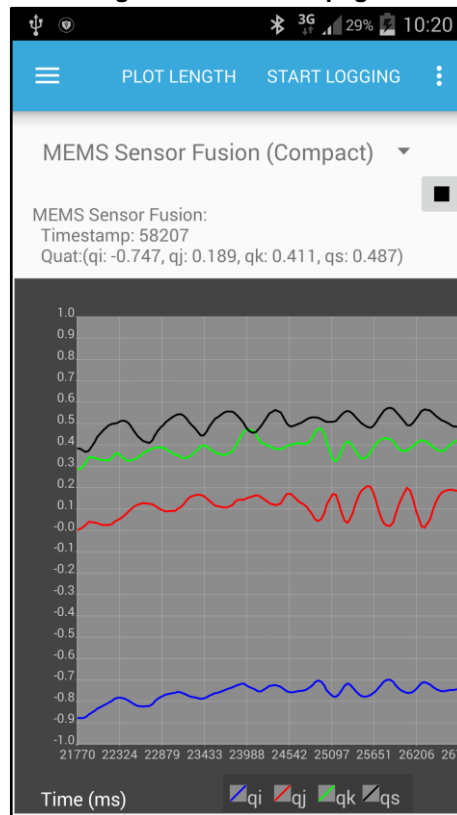
With the default configuration, the application sends three quaternions every 30 ms. When the board is moved, the application harmonizes the sampling rate (the number of quaternions received every second) and the number of rendered frames per second that the smartphone/tablet is capable of processing (60 frames per second is typically the maximum allowed by the GPU). These two values are normally not visible, but can be shown by touching the screen on the top left and on the top right of the display window (only for Android). You can disable and enable the proximity function by pressing the Proximity button.

Figure 8: MEMS Sensor fusion calibration dialog**Figure 9: MEMS Sensor fusion reset board position**

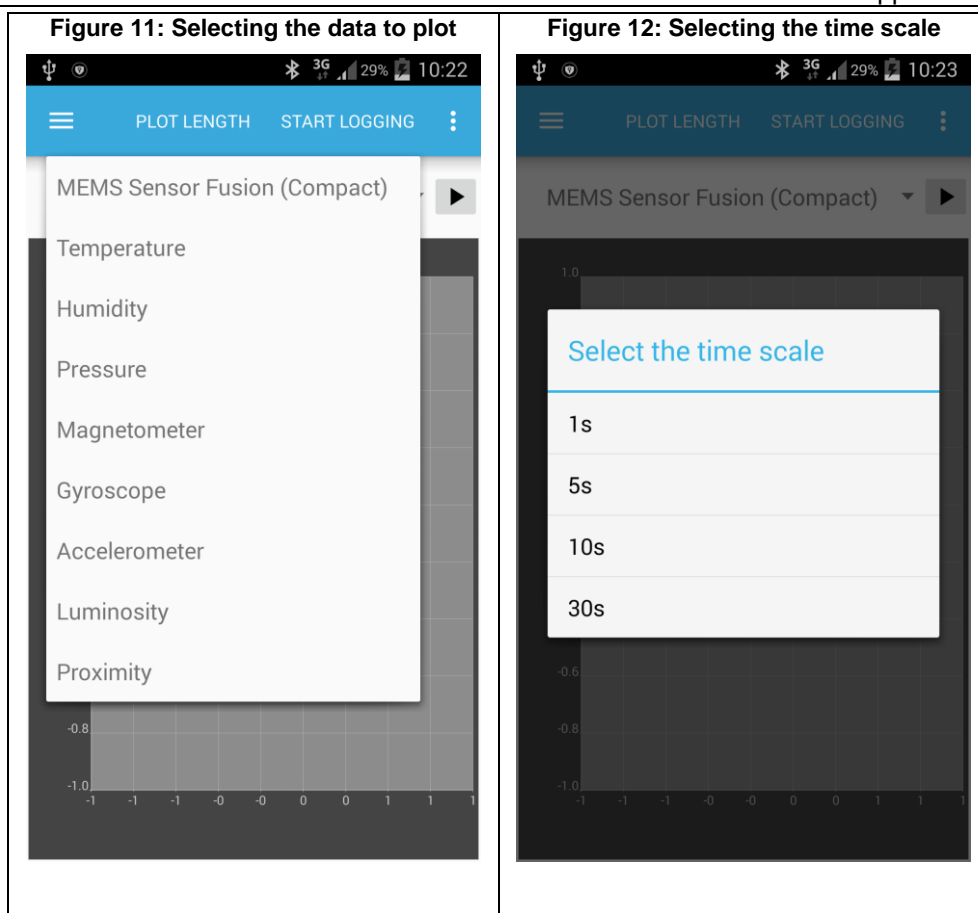
3.1.3 Plot data

In this page, you can plot all the data exported by the device.

Figure 10: Plot Data page



- 1 In order to achieve this, the user has to select the data to plot and press the "Play" button.
By clicking on the Plot length menu item, you can set the time scale in seconds to display.



3.1.4 Activity recognition

This demo requires compatible firmware and you must obtain and load a valid MotionAR license.^a

This page starts the activity recognition algorithm, which can recognize six different activities: standing, walking, fast walking, running, driving and biking.

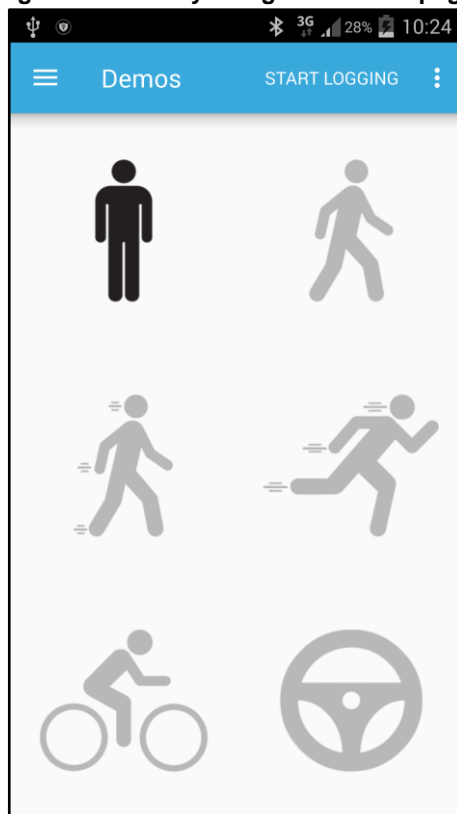
When the algorithm detects a new activity, the associated image turns black. If all the images are grey, the algorithm has not detected any known activities.



As the algorithm must first collect data before recognizing any activities, all the images will be greyed out for a few seconds after the demo starts.

^a See [Section 3.5: "License Manager"](#) to know how to request and load a license from the app.

Figure 13: Activity recognition demo page



3.1.5 Carry position

This demo requires compatible firmware and you must obtain and load a valid MotionCP license.^a

This page starts the carry position recognition algorithm, which detects where the user is carrying the device: on a desk, in hand, near head, in a shirt pocket, in trouser pocket and on swinging arm.

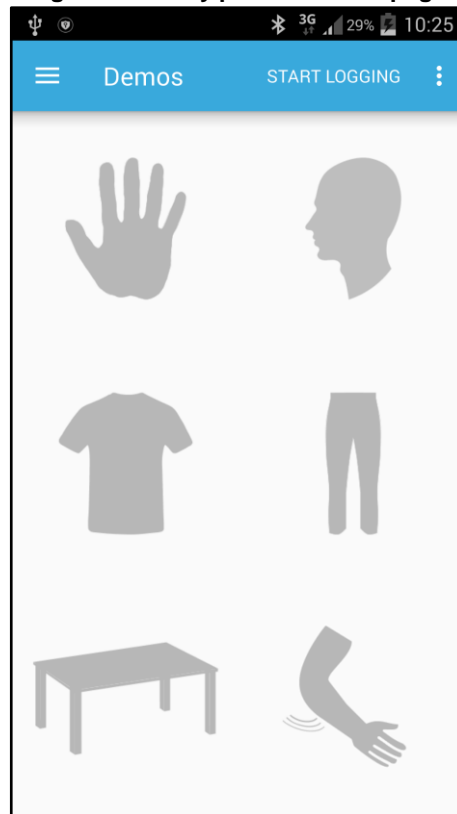
When the algorithm detects a carry position, the corresponding icon turns black. If all the images are grey, the algorithm is detecting a change in the position.



As the algorithm must first collect data before recognizing any activities, all the images will be greyed out for a few seconds after the demo starts.

^a See [Section 3.5: "License Manager"](#) to know how to request and load a license from the app.

Figure 14: Carry position demo page



3.1.6 Gesture recognition

This demo requires compatible firmware and you must obtain and load a valid MotionGR license.^a

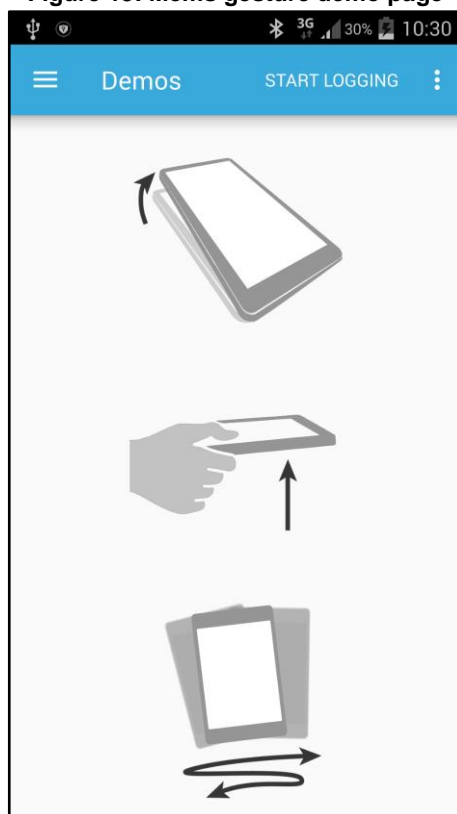
This demo starts the gesture recognition algorithm that uses the information coming from the MEMS sensors to detect certain user gestures:

- Glance: the user moves the device to look at the display (in our case to look at the sensor)
- Pick up: the user picks up the device
- Wake up: the user shakes the device

Each time an event is detected, the icon animates and becomes colored. After three seconds, or when a new event arrives, the icon goes grey again.

^a See [Section 3.5: "License Manager"](#) to know how to request and load a license from the app.

Figure 15: Mems gesture demo page



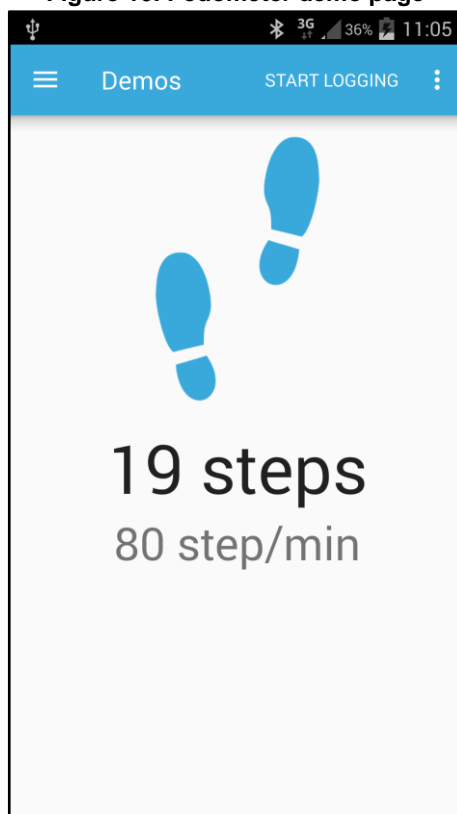
3.1.7 Pedometer

This demo requires compatible firmware and you must obtain and load a valid MotionPM license.^a

This demo starts the pedometer algorithm that uses accelerometer data to count the steps performed by the user and keeps track of the pace in steps per minute.

^a See [Section 3.5: "License Manager"](#) to know how to request and load a license from the app.

Figure 16: Pedometer demo page



3.1.8 Acceleration events

This demo is available when a STEVAL-MKI160V1 board is plugged into the X-NUCLEO-IKS01A1 expansion board or when an X-NUCLEO-IKS01A2 is used. The demo displays the events that are detected by the LSM6DS3/L component (no MCU algorithms involved)

You must select which event to detect; when an event is detected, the corresponding icon shakes.

The available events are:

- Orientation: this event provides the current chip orientation. The related image will show the chip with the dot marker oriented as the real one.
- Free fall: this event is fired when the board hits the bottom after a free fall.
- Single tap: this event is fired when the user taps on the board.
- Double tap: this event is fired when the user performs a double tap on the board in a rapid sequence.
- Wake up: this event is fired when the user moves the board
- Pedometer: this event is fired when a step is detected. The chip is able also to count the number of steps performed after the event detection starts.



The chip does not signal the first six steps

- Tilt: this event is fired when the chip is rotated by an angle bigger than 35 degree for more than 2 seconds. After the event is fired, current position becomes the new reference position.

Figure 17: Orientation event

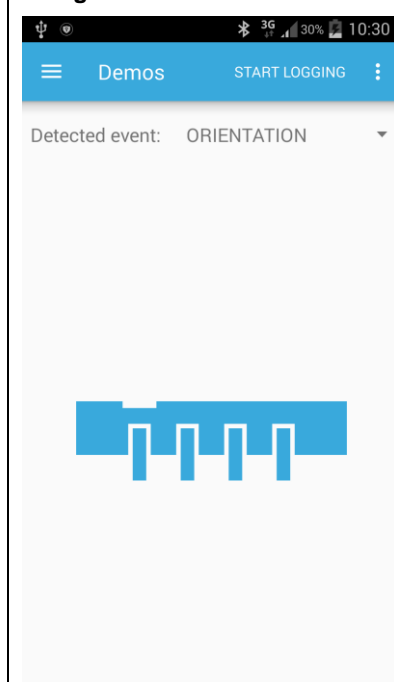


Figure 18: Free fall event



Figure 19: Single Tap event

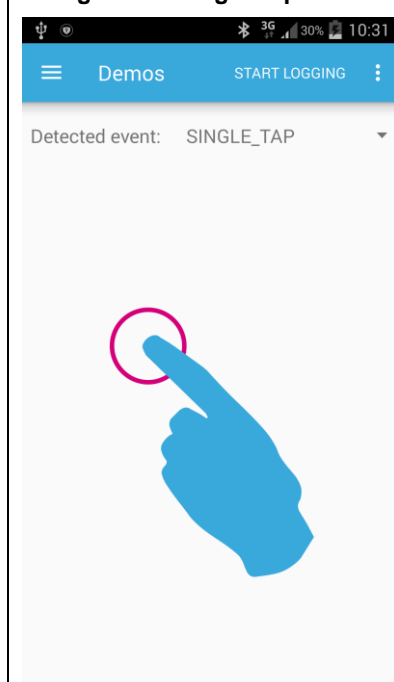


Figure 20: Double Tap event

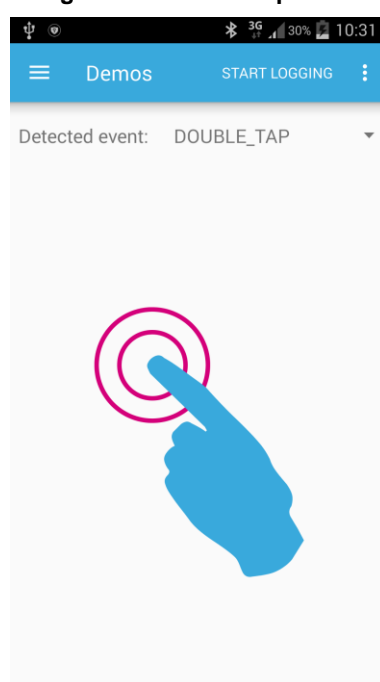


Figure 21: Wake Up event

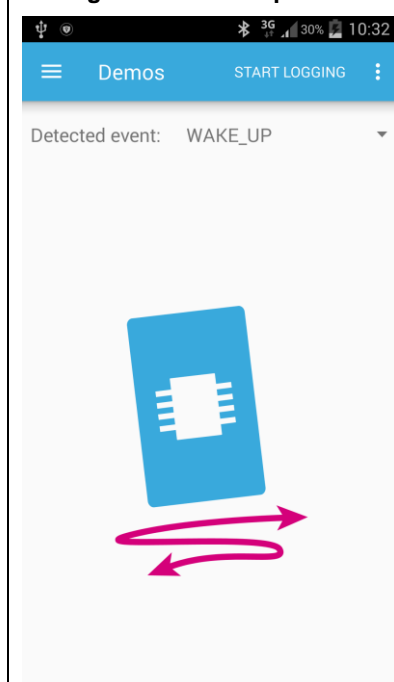


Figure 22: Pedometer event

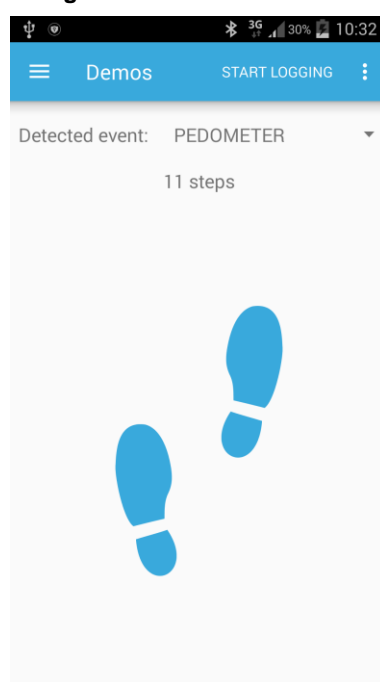
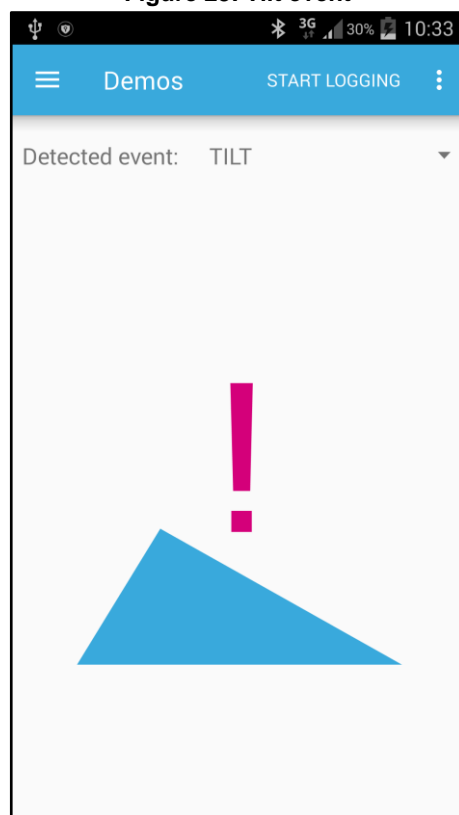


Figure 23: Tilt event

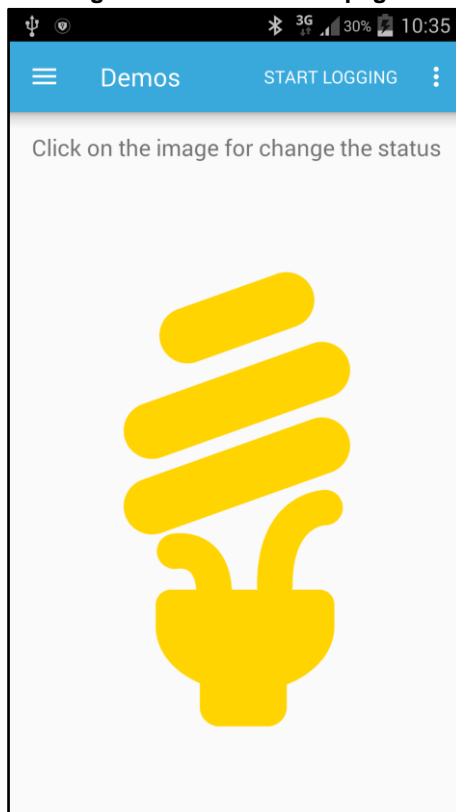


3.1.9 LED switch

This panel can detect and change the status of the user LED on the board.

- 1 Change the status of the LED by clicking on the lamp image.
The image changes according to the action performed only after the board has signaled the new LED status

Figure 24: Switch demo page



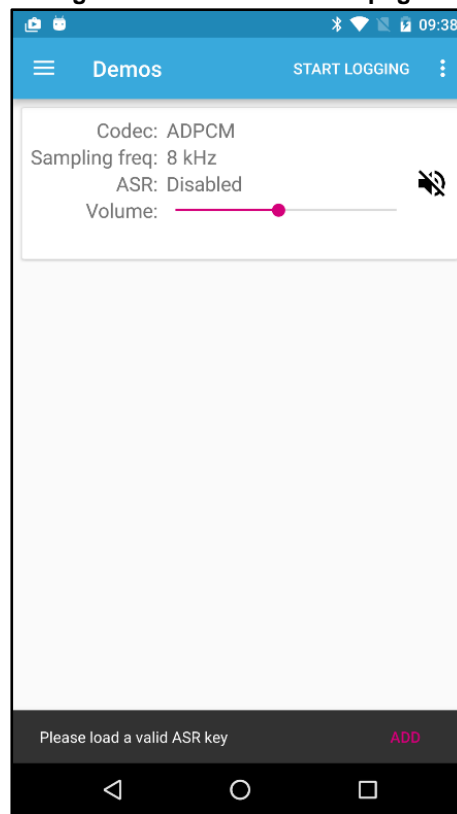
3.1.10 BlueVoice

This demo requires compatible firmware and you must obtain and load a valid BlueVoice license.^a

If the BlueVoice voice over BLE library is enabled, the following page is also available.

^a See [Section 3.5: "License Manager"](#) to know how to request and load a license from the app.

Figure 25: BlueVoice Demo page



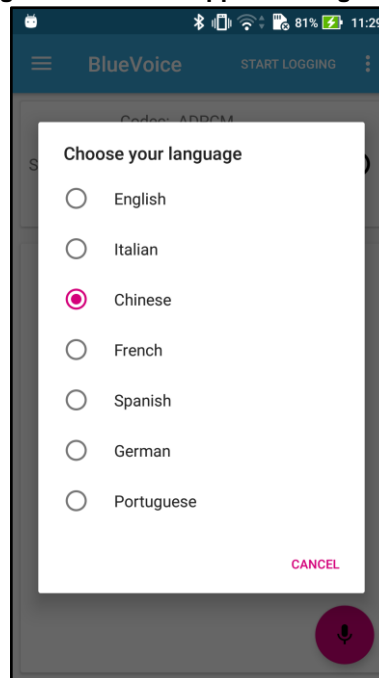
Supported functions are:

- Playback of the audio stream received from the connected device.
- Cloud-hosted speech to text service: iFlyTek services are used for the Chinese language, and Google is used for the other languages.

Audio playback begins as soon as the page is displayed

- 1 Adjust the volume level using the slider or mute by clicking on the speaker icon.
- 2 If the Internet connection is available, the user can use a cloud service to convert the speech to text.
- 3 The app supports different languages and different cloud services. The language can be selected from the “ASR Language” top right menu.

Figure 26: List of supported languages



After selecting the language, the service is initialized, and in case of Google service, a key is asked (refer to [Section 4: "Google speech ASR Key generation"](#) for details on how to obtain the key). After inserting a valid key, the microphone button appears and the speech-to-text service is shown.

Figure 27: Add the service key request

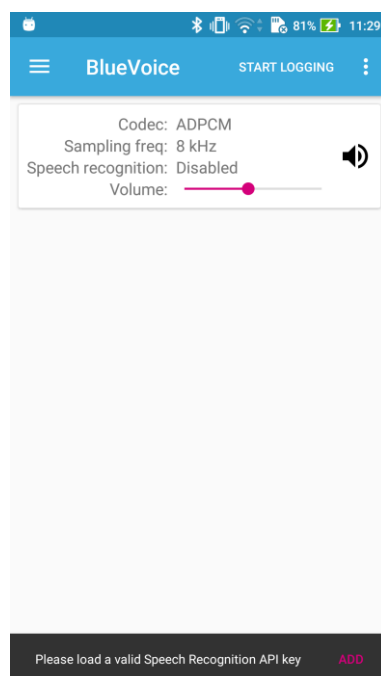
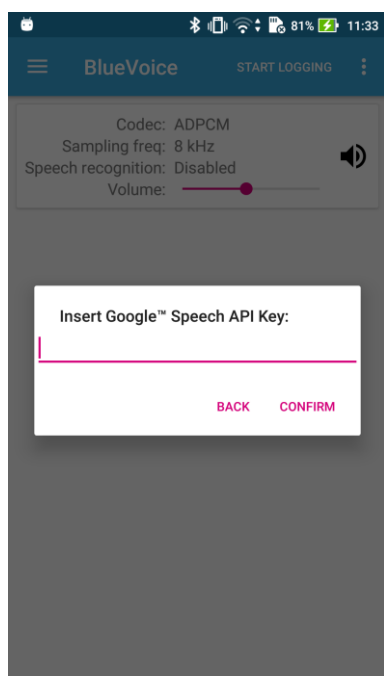
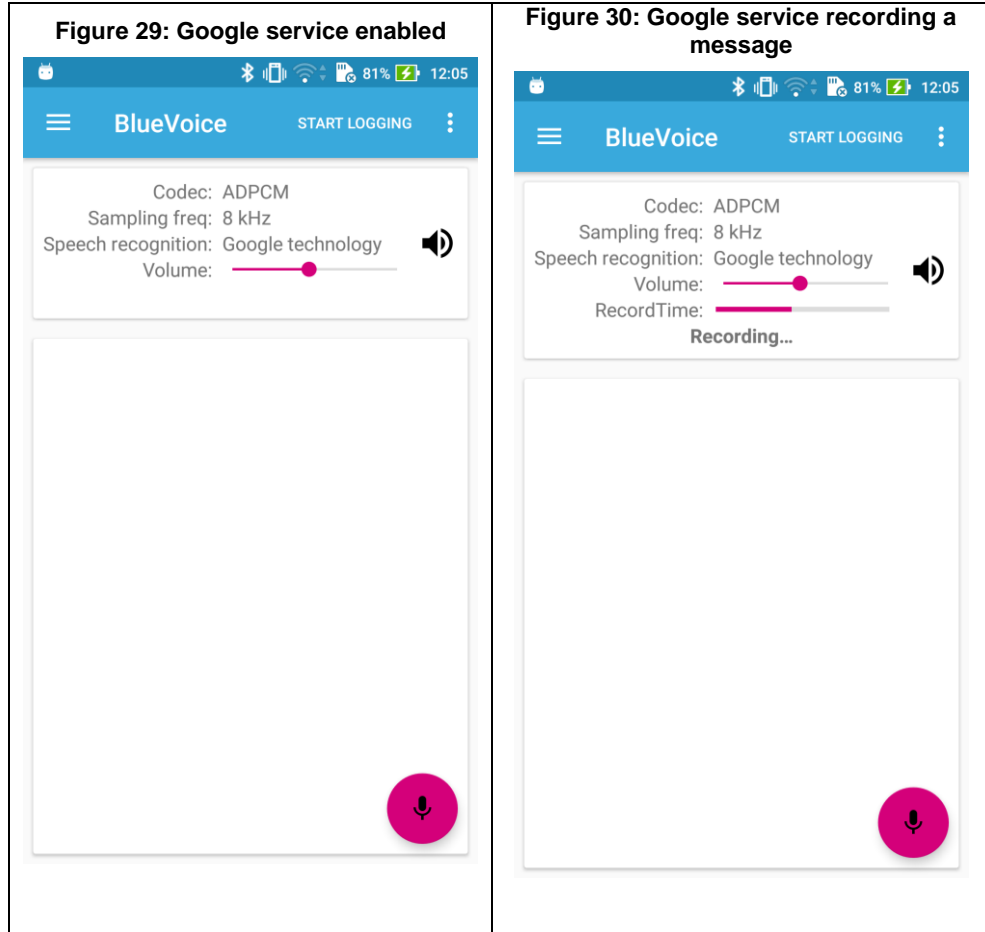


Figure 28: Insert key dialog box



- 4 The Google service works as a walkie-talkie: keeping the microphone button pressed, the audio is recorded (for a maximum of 5 seconds) and sent to the service as soon as the button is released.
- The Google service returns also a confidence result. The app will show a recognized text only if confidence is bigger than 75%, otherwise an error message is shown.



- 5 iFlytek service works in streaming mode: the audio is sent as a real-time stream to the cloud service.
- To start recognizing the audio, the user can press the microphone button, which becomes green as long as the app acquires the audio and the cloud service recognizes the speech. Pressing the microphone button again stops the streaming to the service.

Figure 31: iFlytek service enabled

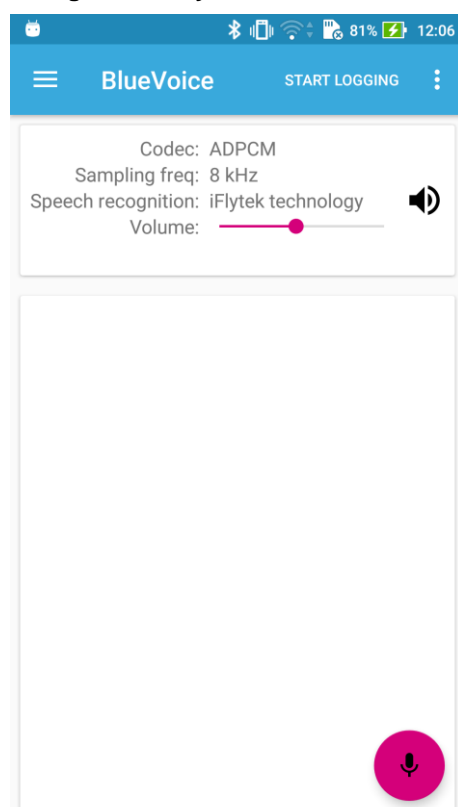
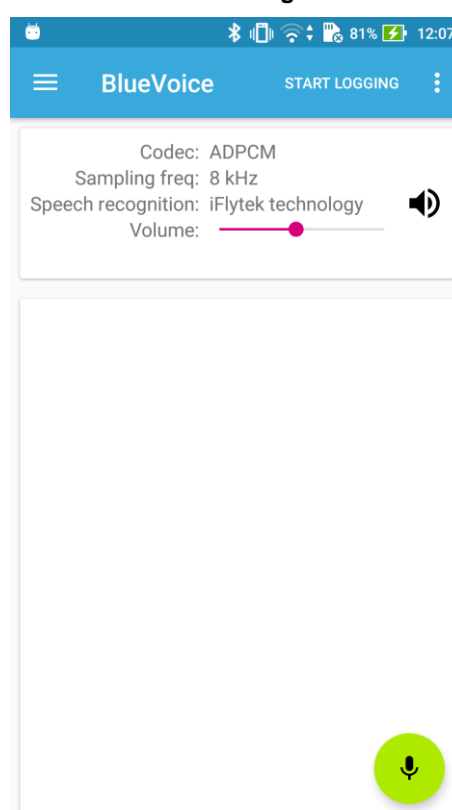


Figure 32: iFlytek demo during the streaming



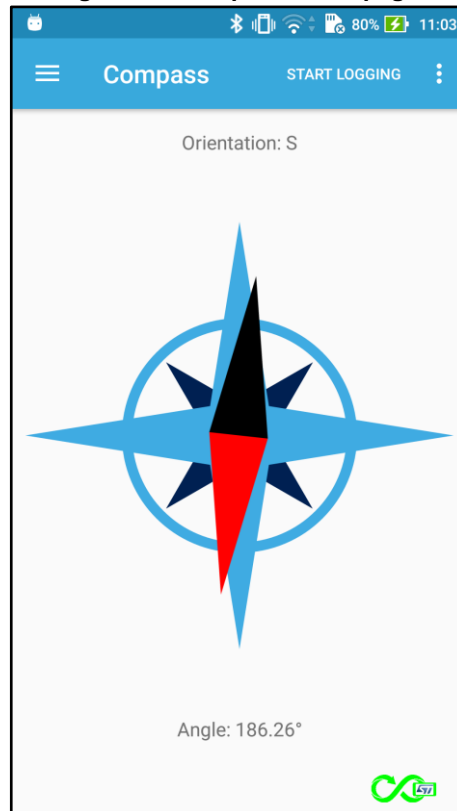
3.1.11 Compass

This demo is shown when the MotionFx library is available in the system, since the compass angle is directly computed by the library.

The demo shows the board orientation with respect to the magnetic north.

To have a correct and stable result it is necessary to calibrate the onboard magnetometer. The calibration process will start automatically if the board is not already calibrated.

Figure 33: Compass demo page



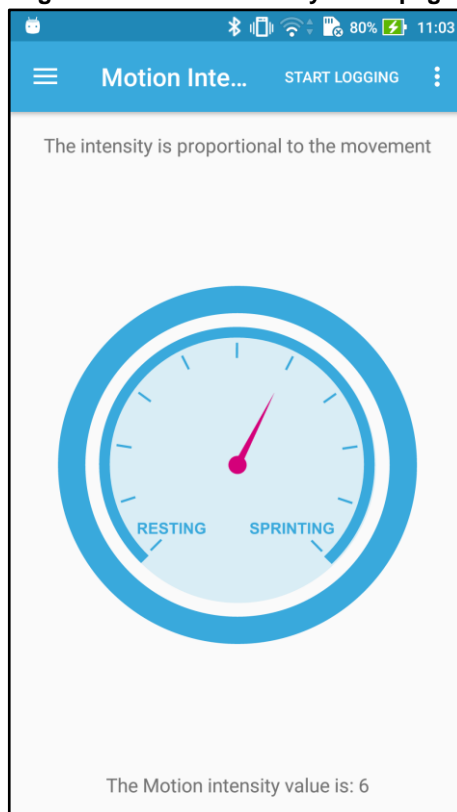
3.1.12 Motion intensity

To enable this demo a compatible firmware is needed and user must obtain and load a valid MotionID license.^a

The library analyzes the motion sensor data and classifies the intensity of the motion in a range from 0 to 10, where 0 is the absence of motion and 10 is equivalent to sprinting.

The demo shows this number in a speedometer.

Figure 34: Motion intensity demo page

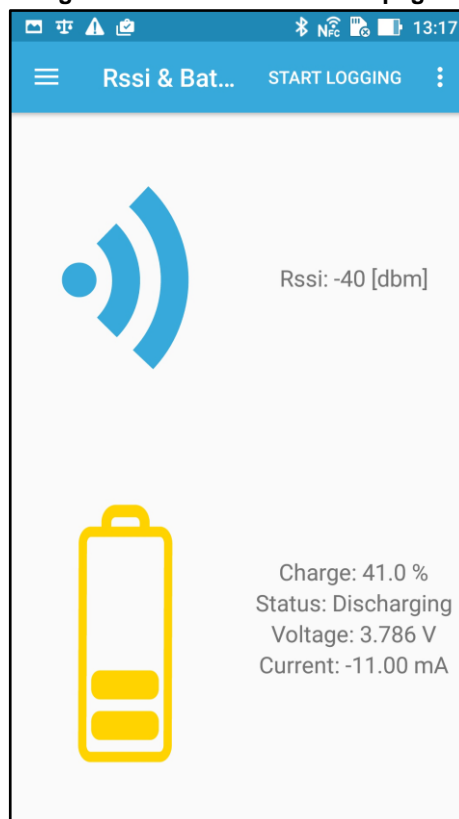


^a See [Section 3.5: "License Manager"](#) to know how to request and load a license from the app.

3.1.13 Rssi and battery

This page shows RSSI of the Bluetooth signal strength and, for STEVAL-STLKT01V1, if the battery is connected, the charge percentage, measured voltage and battery status (charging/discharging/low battery).

Figure 35: Board status demo page



The RSSI value is updated every second.

3.1.14 Cloud

This demo allows sending the sensor data to a cloud provider and requires a mobile phone with a working Internet connection (otherwise, it remains disabled).

The application supports the IBM Watson IoT platform which can be used in two modes:

- through the “Quickstart” registration-free IBM application that shows the data in a plotted graph;
- via a registered IBM Watson account, that enables pushing the data to a custom cloud application.

3.1.14.1 IBM Watson Quickstart

Figure 36: Data needed for the Quickstart service

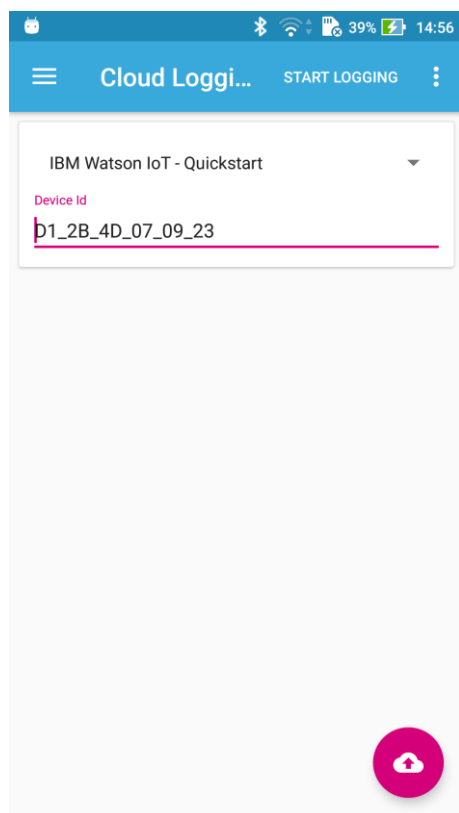
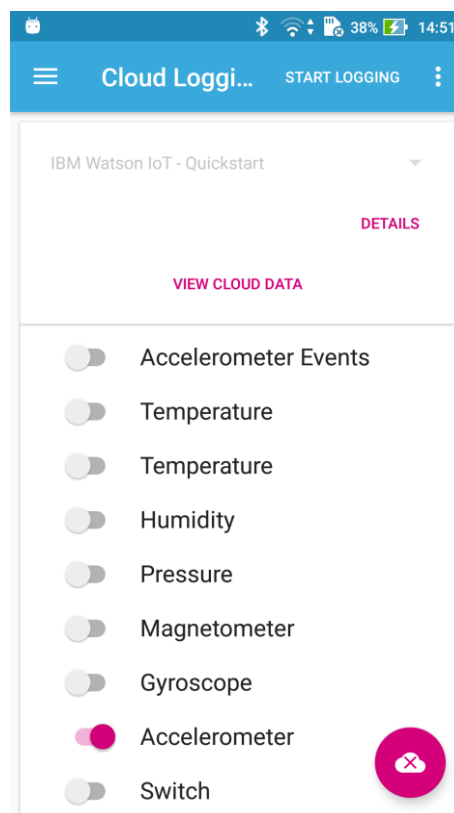


Figure 37: List of available features after the connection



To use the “Quickstart” mode, select the “IBM Watson – Quickstart” option from the menu.

The app requires then the insertion of the “Device ID” string, which is the name to display the data in the web page. The Device ID must be unique and the default value is the board MAC address.

As soon as the connection with the cloud has been established, a list of the available features is shown. To push a specific data to the cloud, the corresponding switch must be turned on.

ST BlueMS “Quickstart” mode only allows to push one sample per second to not overload the webpage.

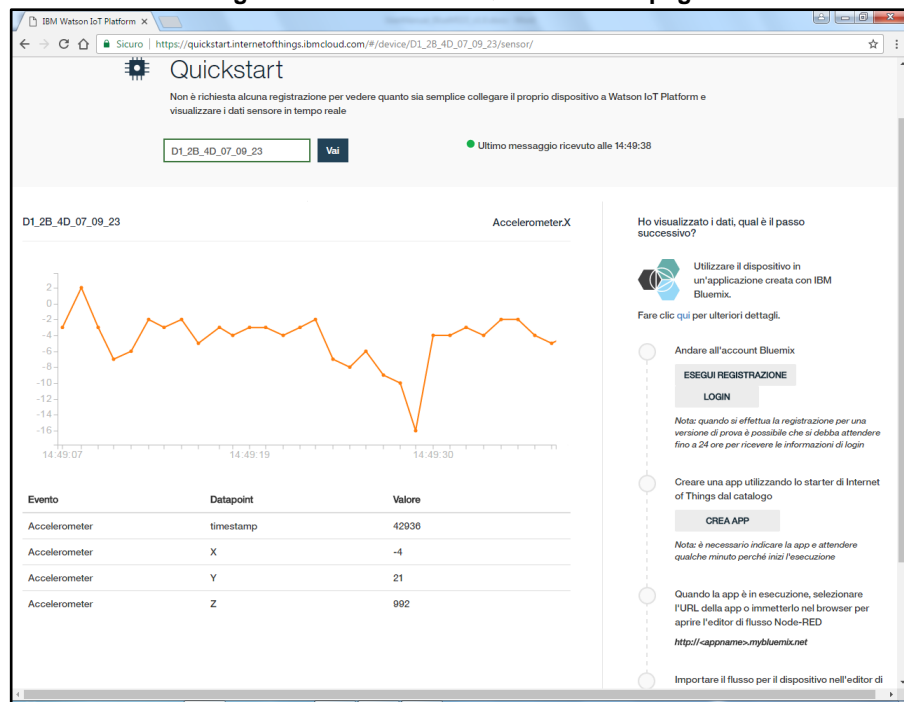
To see the cloud data, the user can either:

- click on the “View Cloud Data” button and open the webpage from the mobile phone
- or visit the page <https://quickstart.internetofthings.ibmcloud.com>.



In the webpage the user must accept the terms of service, insert the “Device ID” and press the “Go” button to show the plot.

Figure 38: IBM Watson Quickstart webpage



3.1.14.2 IBM Watson registered account

To use the IBM Watson platform, select "IBM Watson" option from the menu.

Before connection starts, the following information must be entered:

- organization ID: the name of the organization that is registered in the IBM platform
- authentication token: used to open a secure connection with the IBM servers
- device type: the device type used
- device ID: a unique ID for the device



All fields are mandatory.

To show the cloud data click on the "View Cloud Data" button and open the webpage from the mobile phone or visit the page:

[https://\[organization\].internetofthings.ibmcloud.com/dashboard](https://[organization].internetofthings.ibmcloud.com/dashboard).

It is also possible to use a demo ("play") account (refer to [Section 5: "IBM Watson IoT"](#) for further details).

3.2 Secure pairing

Some firmware versions require the insertion of a six-digit pin to complete the Bluetooth connection. The pin is generated randomly every time the board starts.

You can obtain the pin from:

1. the serial console
2. the digits '00' plus the digits on the X-NUCLEO-6180XA1 board display



Note the pin before starting the connection procedure as the display is switched off when the connection begins.

After the first connection, the device is recognized as trusted and the pin is no longer necessary.

3.2.1 NFC

Android platforms have setting options which allow the use of NFC tag connectivity to automatically launch applications and initiate connections.

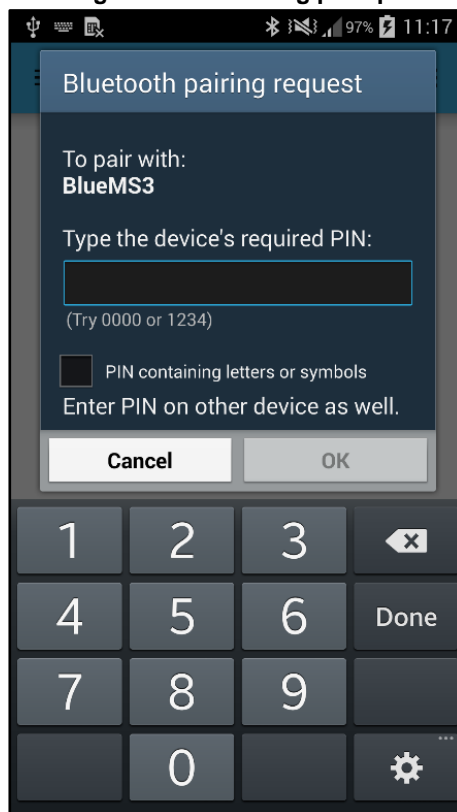
When a properly configured Android device reads the NFC tag, the BlueMS application automatically opens the user PIN prompt, inserts the right PIN and proceeds with device pairing.

With Android version 5.0 and above, the system may show two messages: the first requesting confirmation to begin and the second prompting for the pin (which you enter only if you are not using the NFC tag functionality).



The messages do not always appear in the foreground; you may have to access them via the Notification Center.

Figure 39: PIN dialog prompt



3.3 Debug console

Clicking “Open Debug Console” in the overflow menu launches the debug console, which is a serial console able to send commands to and receive messages from the board.

The default implementation of this service simply echos the commands entered by the user.

The strings written by the user are displayed in blue, the data returned by the board are black and the error messages are red.

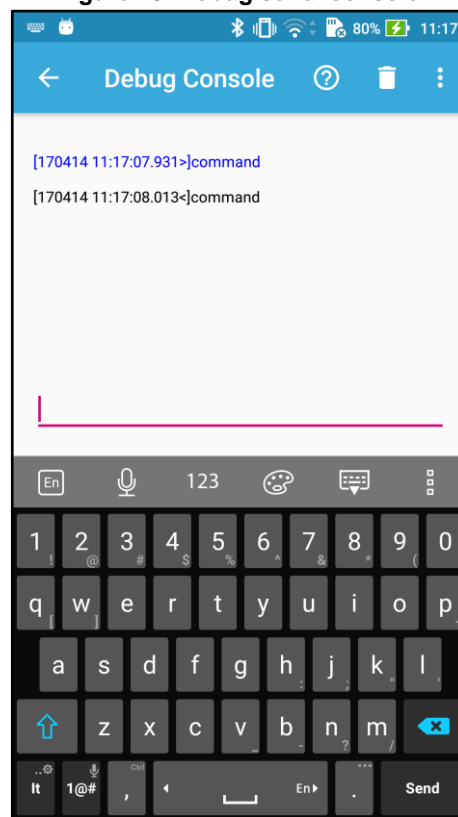


Communication is managed by the Bluetooth low energy protocol, so commands longer than 20 bytes are split into multiple messages over the communication channel.

A click on the question mark button sends the “help” command and shows the list of available commands.

Clicking on the trash can button will clear the console history.

Figure 40: Debug serial console



3.4 Log settings

On Android platforms, you can open the setting menu to change format and location of stored log data.

The setting menu lets you:

- change the path where the logs are stored

- change how the data is stored: CSV by default, but you can use a database format or print them to the Android console (LogCat)
- remove the previously generated files.

Figure 41: Settings screen

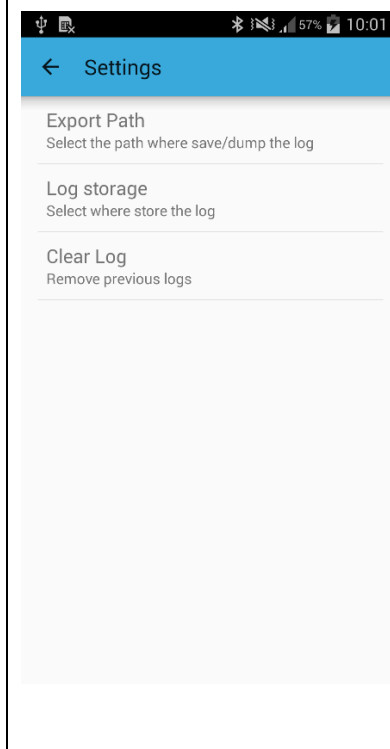


Figure 42: Path to save the log data

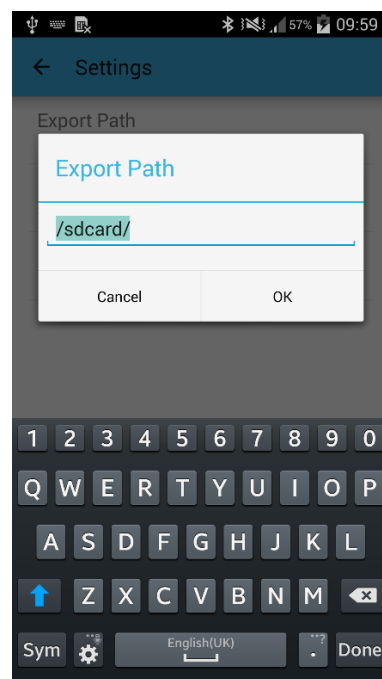
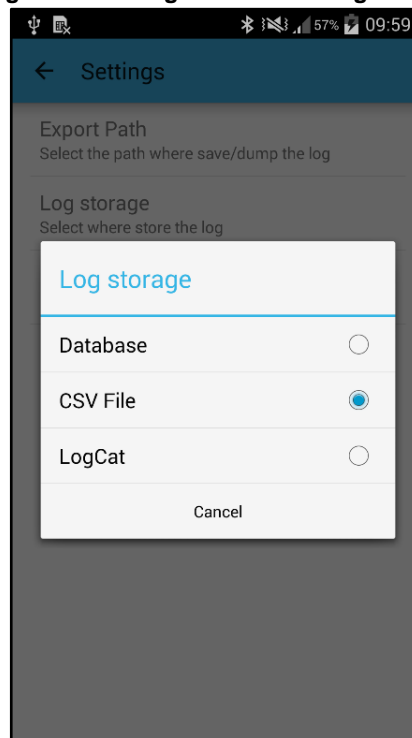


Figure 43: Dialog to select the log format



3.5 License Manager

For the functions requiring a valid license, requests must be made for each STM32 Nucleo board where the firmware will be flashed.

This latest version of BlueMS lets you request licenses directly from your smartphone and upload the license code into the firmware via BLE, so you do not have to recompile the firmware for each node.

To access this functionality, select the License Manager from the menu in the demo view.

Figure 44: Main License Manager view

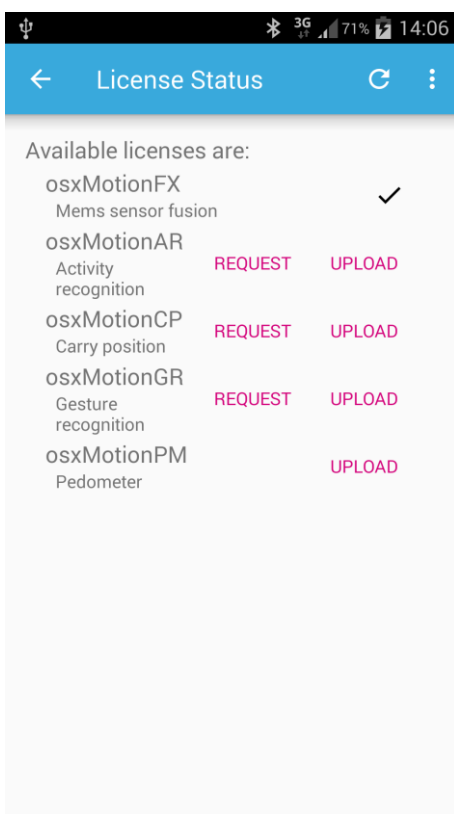
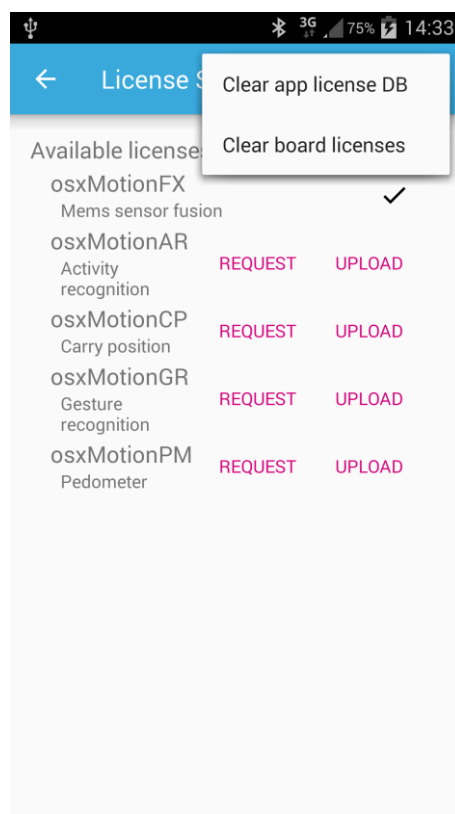


Figure 45: License Manager menu



If the firmware supports the License Manager, the application lists the available licenses and their status.

From this view, you can:

- request and upload a license
- clear app license DB: the app stores all the licenses already uploaded to avoid requesting them again; this button removes all the licenses stored in the app.
- clear board license: this deletes all the licenses on the board; this also resets the board, so you need to restart BlueMS as well.

3.5.1 Requesting a license

BlueMS keeps a copy of licenses that have already been requested so you do not have to request them again.

- 1 Click "Request" to request a new license.
- 2 Read the license agreement and then click the "Agree" button if you wish to proceed.
- 3 You will be taken to a new view where you need to fill in the relevant information.



Avoid using non-Latin characters.

User data is saved by the application so you do not have to enter it again for subsequent licenses.

- 4 Once all the fields are compiled, the user can press the "Send" button. The app will then generate the request e-mail and open the default e-mail client to send the e-mail to the ST server. In few minutes the user will receive an e-mail with the license code.

Figure 46: Example of license to agree with

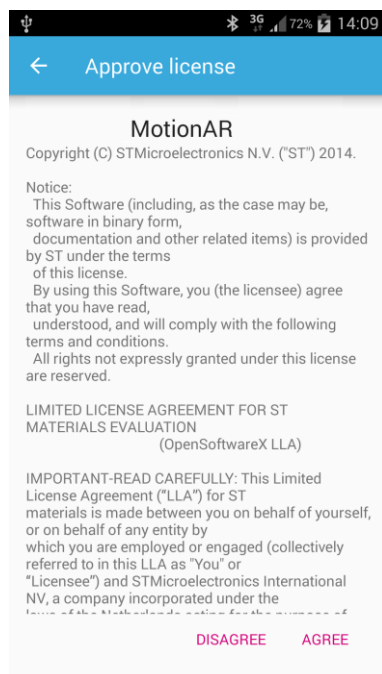
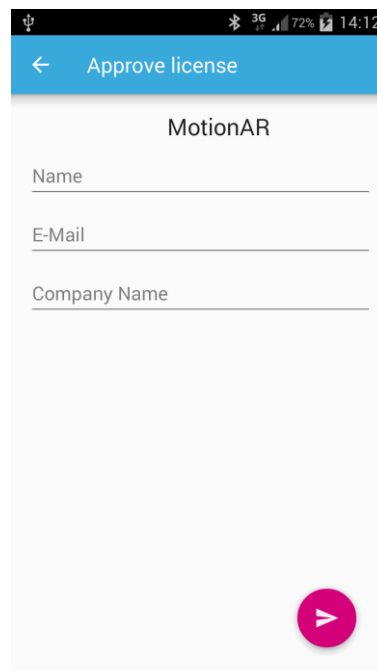
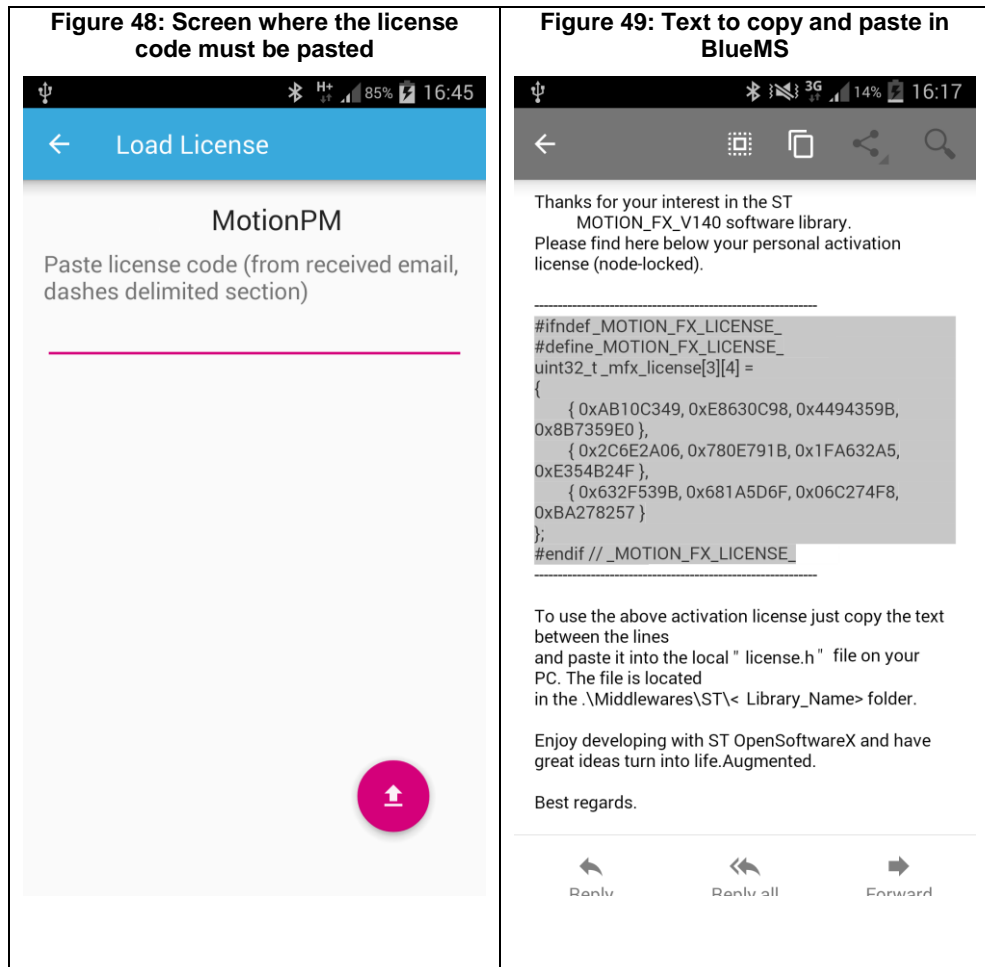


Figure 47: Fields to compile to request the license



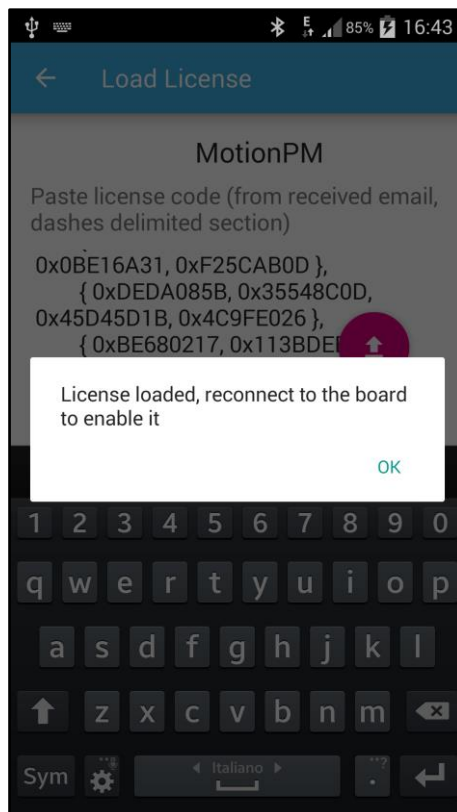
3.5.2 Uploading the license

- 1 Press the "Upload" button to upload the license to the device.
Locally stored licenses will be directly loaded to the board, otherwise you will be prompted to paste the license activation code supplied in the e-mail reply.



- 2 A dialog will inform the user whether the license has been correctly uploaded and accepted by the system.

Figure 50: License loaded confirmation message



- 3 After the license has been accepted, reconnect to the device to apply the changes. If the license is rejected, the board must be reset. Whenever a license is correctly loaded onto a board, it is also saved by the application for reuse when, for example, the board is re-flashed.

3.6 Firmware upgrade

If the firmware you are running supports it, you can update the firmware from BlueMS. The firmware can be retrieved locally (Android only) or from a cloud storage service like iCloud®, Google Drive® or Drobpox®.

- 1 Click on "Firmware upgrade" from the menu in the demo view to start the firmware upgrade procedure. The main view will show information regarding the current firmware version.

Figure 51: Firmware upgrade main view



- 2 Click on the “Upload” button and select the binary file to upload.
The upload process begins immediately.



The binary file must only contain the application firmware; it is not possible to upgrade the boot loader as well.

The Flash is wiped and the new firmware is flashed.

Figure 52: Formatting the flash memory

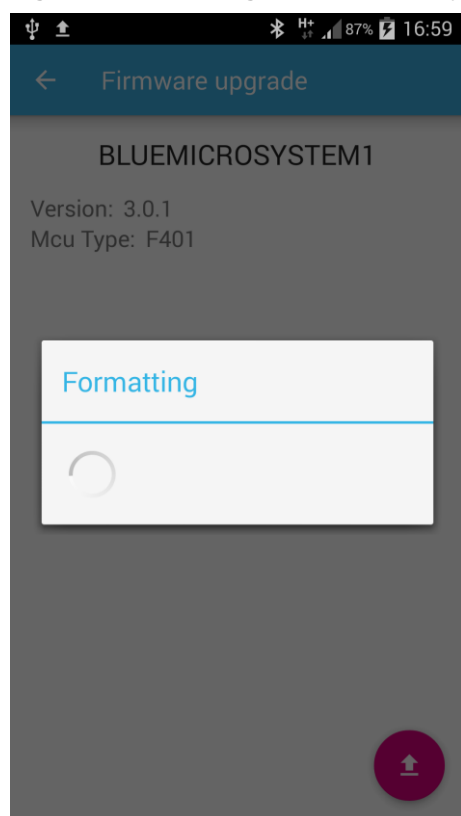
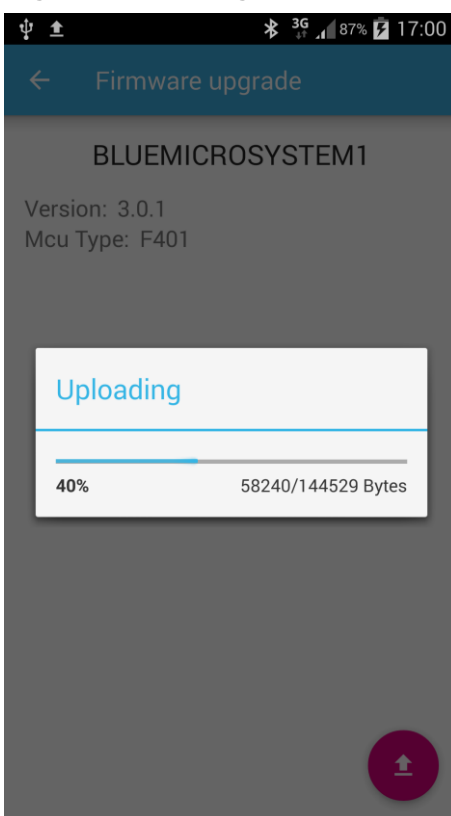
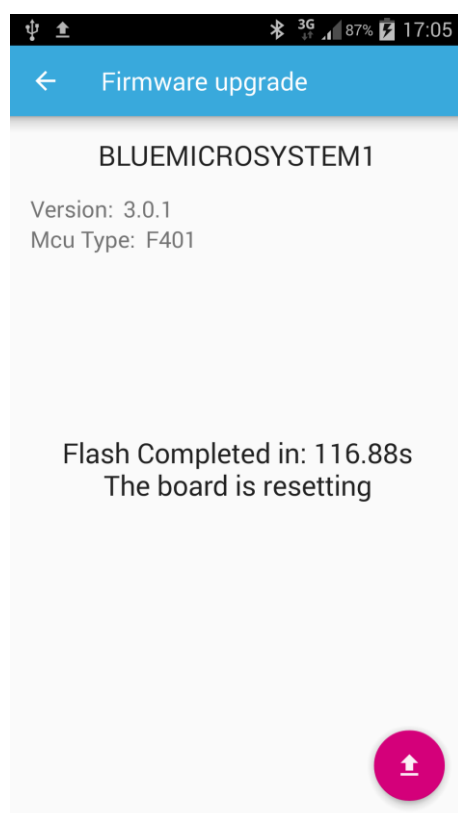
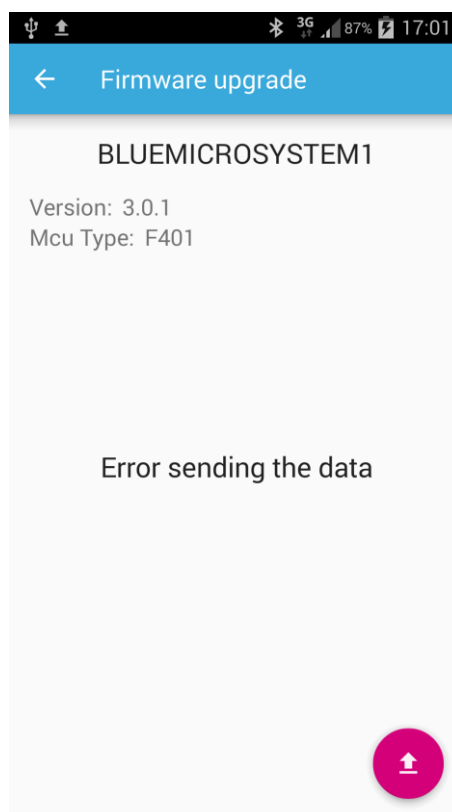


Figure 53: Uploading the new firmware



- 3 Once the new firmware is correctly uploaded to the board, the board will restart.
- 4 Restart the BlueMS application.

Figure 54: Uploading complete with success**Figure 55: Error during the uploading**

- 5 Some Android devices do not support the default transmission rate used by the application to upgrade the firmware over BLE, which may lead to transmission errors.
- 6 In this case:
- reset the board
 - wait for the application to acknowledge the disconnection
 - reconnect with the board and try to upload the firmware again.

The upload rate should now be slower, thus reducing the probability of transmission error. In case of error, repeat the procedure to further reduce the transmission rate.

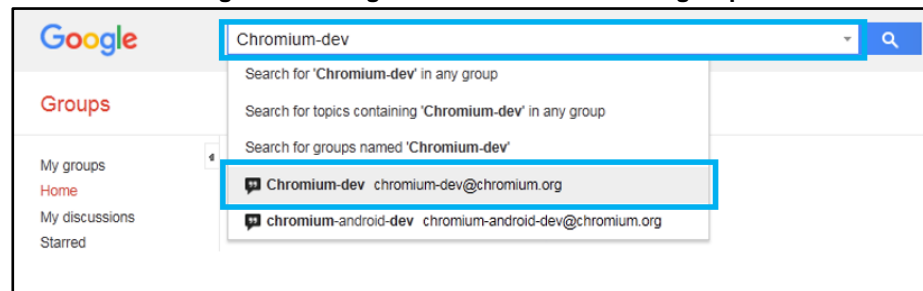
4 Google speech ASR Key generation

The Google Speech APIs require a key to access the web-based service. You need a Google account to complete the procedure and access the service.

To generate a key:

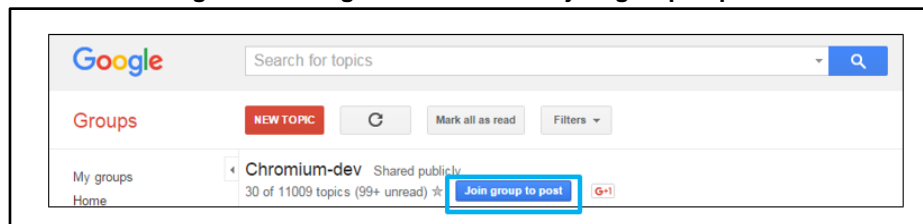
- 1 Login with your own Google account.
- 2 Subscribe to Chromium-dev at
<https://groups.google.com/a/chromium.org/forum/?fromgroups#!forum/chromium-dev>.
- 3 Write "Chromium-dev" in the search box, and select the appropriate group.

Figure 56: Google Chromium-dev: search group



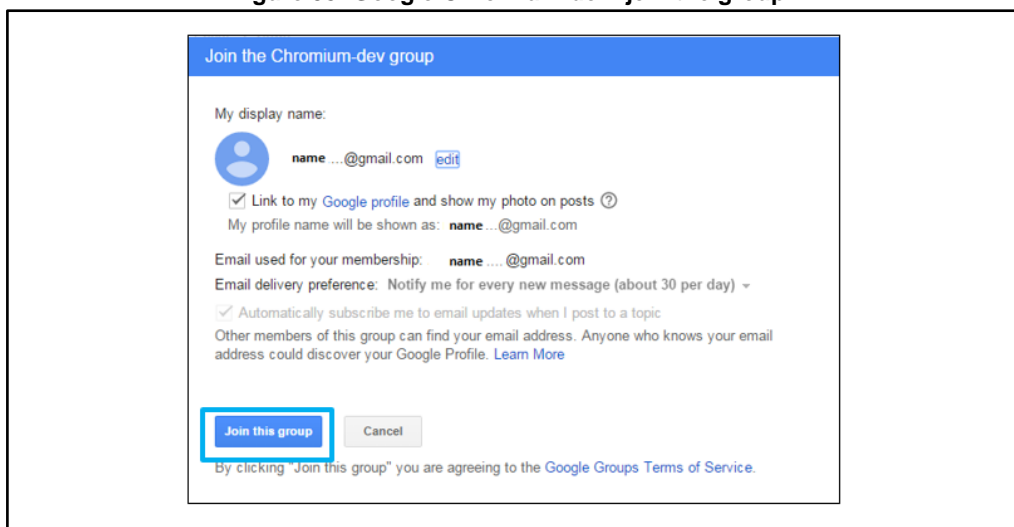
- 4 Click on "Join group to post" button

Figure 57: Google Chromium-dev: join group to post

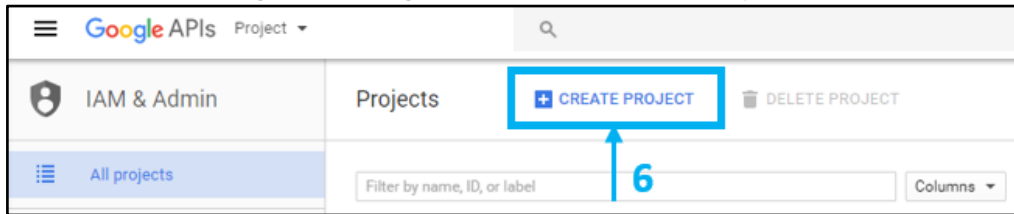


- 5 Click on "Join this group" button to join the Chromium-dev group.

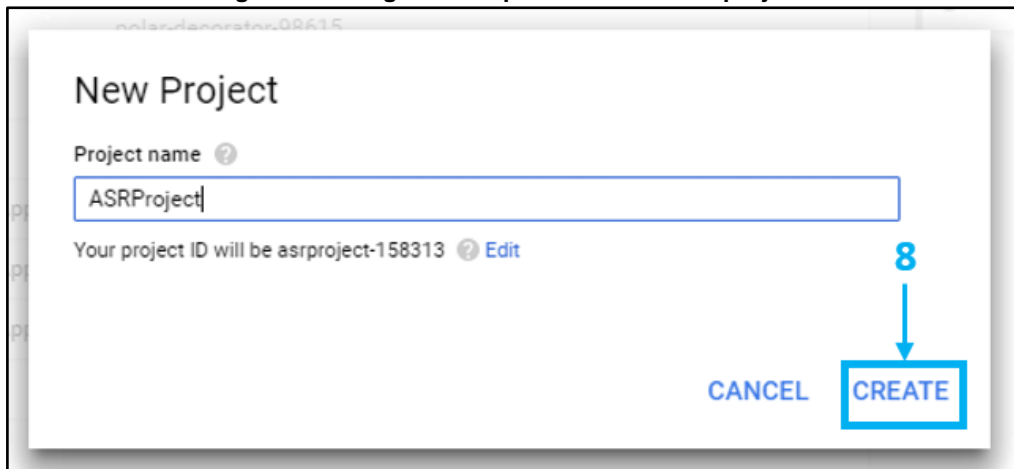
Figure 58: Google Chromium-dev: join the group



- 6 Go to <https://console.developers.google.com/project>
- 7 Click on "Create a project..."

Figure 59: Google Chromium-dev: create project

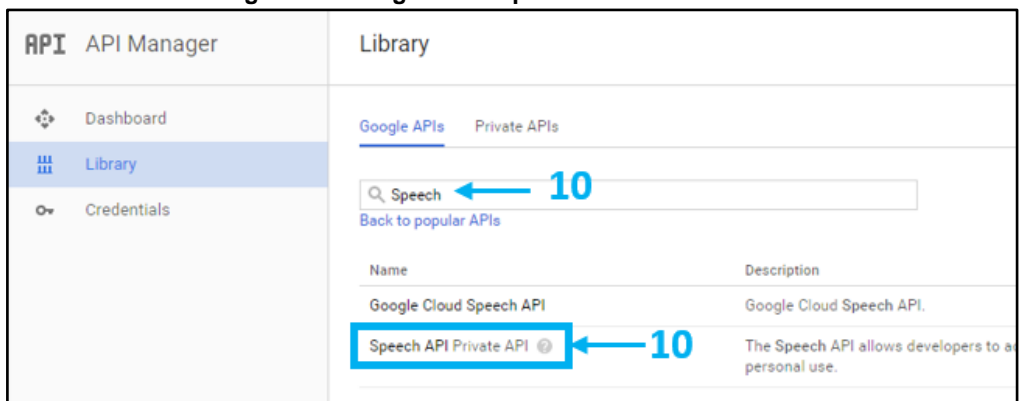
- 8 Choose the Project name.
- 9 Click on "Create" button.

Figure 60: Google Developers Console: new project

- 10 Make sure you have selected the newly created project.

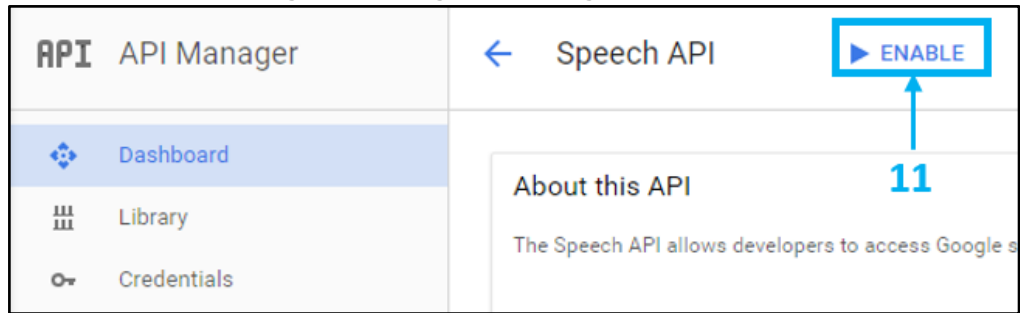
Figure 61: Google Developers Console: ASRProject

- 11 Write "Speech API" in the search box, and select correct result.

Figure 62: Google Developers Console: select API

- 12 Enable the Speech API clicking on the blue button.

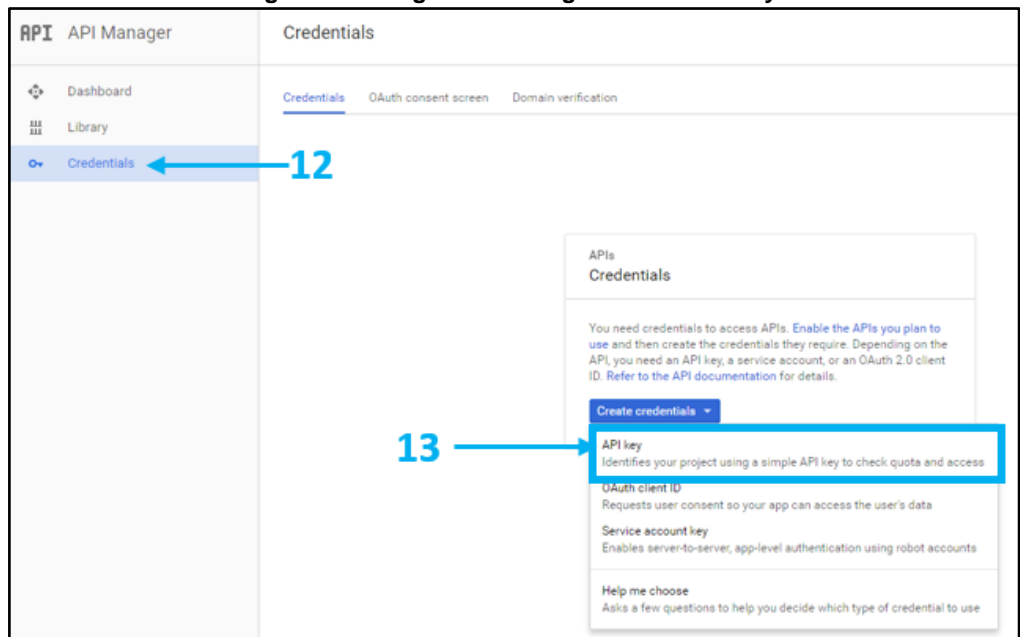
Figure 63: Google API Manager: enable API



- 13 Move from the “Dashboard” tab to “Credentials” tab.

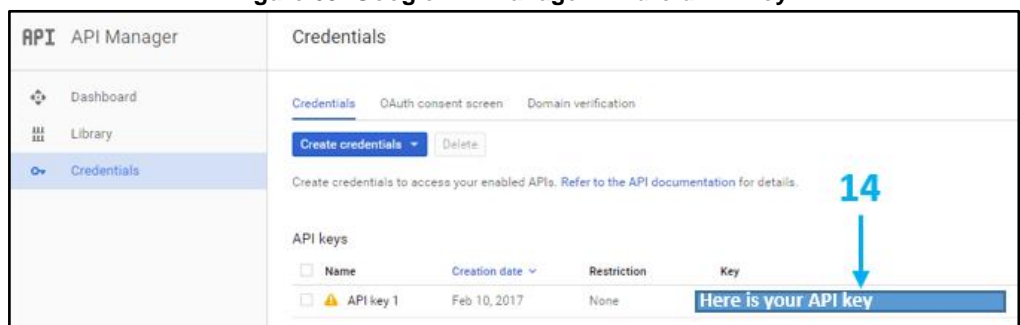
- 14 Open the “Create credentials” menu and select “API key”.

Figure 64: Google API Manager: create API key



- 15 Your API key is created. Click on Close to return to the Credentials section. Here you can see your API Key.

Figure 65: Google API Manager: Android API key



5 IBM Watson IoT

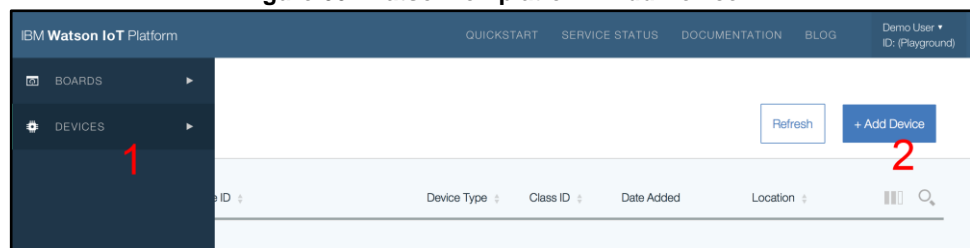
5.1 Register a device

It is possible to register a device in demo mode to test IBM cloud services.

To register the device:

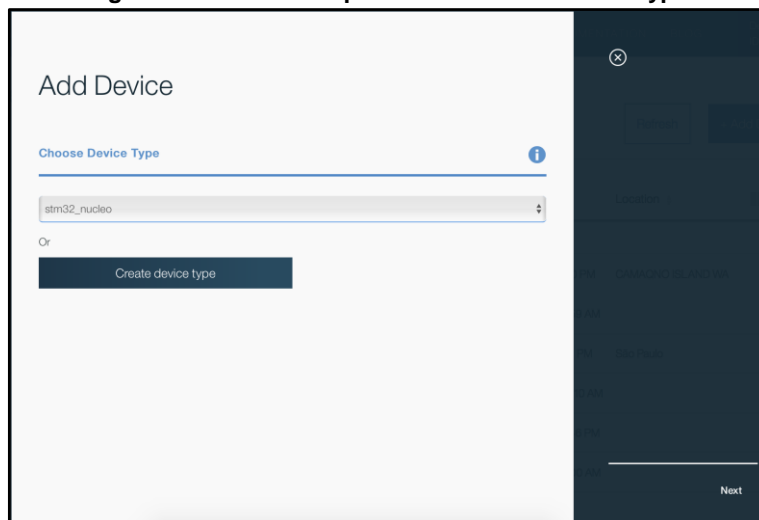
- 1 Visit <https://play.internetofthings.ibmcloud.com/dashboard/>
- 2 Click on "Add Device"

Figure 66: Watson IoT platform: Add Device



- 3 Choose a device type.
You can create your own type or use the existing "stm32_nucleo".

Figure 67: Watson IoT platform: Choose Device Type



- 4 Choose the device ID (device name).
- 5 Choose an authentication token (a random one can be used).



Take note of the token as it is not possible to recover it later.

- 6 Confirm all the data.

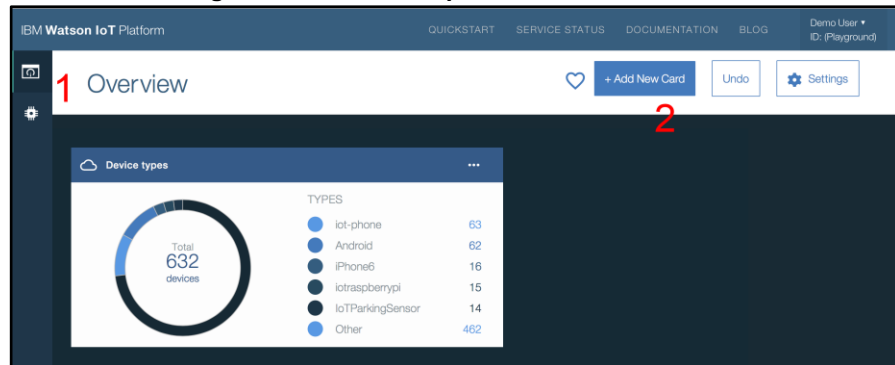
5.2 Display data

When using the BlueMS app, the user should enter the device ID/type, the authentication token, and “play” as the organization name in the proper fields.

To display the board:

- 1 Visit <https://play.internetofthings.ibmcloud.com/dashboard/>
- 2 Open or create a new board configuration.
- 3 Add a new card.

Figure 68: Watson IoT platform: Add New Card



- 4 Select the linear plot card .
- 5 Search the related device ID and select it.
- 6 Add a new dataset that will be displayed in the plot.
If the board is already sending data, the user can choose the event and property from the menu.

Figure 69: Watson IoT platform: add a new dataset

The screenshot shows the 'Create Line chart Card' dialog. The left sidebar has three sections: 'Card source data' (with 'bluemsTest' selected), 'Card preview', and 'Card information'. The main area is titled 'Create Line chart Card' and contains a 'Connect data set' section. Below this, there are input fields for 'Event' (containing 'X'), 'Property' (containing 'Accelerometer'), 'Name' (containing 'X'), 'Type' (containing 'Text'), and 'Unit'. A 'Connect new data set' button is at the bottom left.

- 7 Press “Next”.
From this moment the user can see in real-time the data coming from the node.
The plot type, size, name and color can also be changed.

6 Revision history

Table 3: Document revision history

Date	Version	Changes
09-Dec-2015	1	Initial release.
10-Oct-2016	2	<p>Throughout document:</p> <ul style="list-style-type: none"> - text and formatting changes - updated all images to reflect new version interface - added osxMotionGR and osxMotionPM application support information <p>Added <i>Section 3.1.6: "Gesture recognition"</i></p> <p>Added <i>Section 3.1.7: "Pedometer"</i></p> <p>Added <i>Section 3.5: "License Manager"</i></p> <p>Added <i>Section 3.6: "Firmware upgrade"</i></p> <p>Added <i>Section 3.1.8: "Acceleration events"</i></p> <p>Added <i>Section 3.1.9: "LED switch"</i></p> <p>Added <i>Section 3.1.10: "BlueVoice"</i></p> <p>Added <i>Section 4: "Google speech ASR Key generation"</i></p>
30-May-2017	3	<p>To reflect BlueMS 3.4 version:</p> <ul style="list-style-type: none"> - updated <i>Section "Introduction"</i>, <i>Section 2: "Setup"</i>, <i>Section 3.1.10: "BlueVoice"</i>, <i>Section 3.3: "Debug console"</i> and <i>Section 4: "Google speech ASR Key generation"</i>; - added <i>Section 3.1.11: "Compass"</i>, <i>Section 3.1.12: "Motion intensity"</i>, <i>Section 3.1.14: "Cloud"</i>, <i>Section 3.1.14.1: "IBM Watson Quickstart"</i>, <i>Section 3.1.14.2: "IBM Watson registered account"</i>, <i>Section 5.1: "Register a device"</i> and <i>Section 5.2: "Display data"</i>.

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.

© 2017 STMicroelectronics – All rights reserved