Introduction

STBLESensor application is available for Android and iOS and shows the data exported by a BLE device using the BlueST protocol.

The app shows different panels based on the data types exported by the firmware, including: environmental data, MEMS sensor fusion, plot, activity recognition, carry position, acceleration event, BlueVoice, speech to text, beamforming, sound source localization, pedometer, switch, motion intensity, compass, cloud logging, node status.

All the data received by the app can be logged in CSV files and exported by e-mail.

If the firmware supports the functionality, the application can also show a serial console to exchange string messages with the board. This functionality is also used to upgrade the board firmware.

Both Android and iOS applications support the Bluetooth secure pairing with pin.

For Android it is also possible to use an NFC tag to read the pin and trigger the connection with the node.
1 Overview

The STBLESensor mobile application has been developed to support expansion software packages based on the BlueST-SDK BLE protocol (version 1.0 or above of FP-SNS-ALLMEMS1, FP-SNS-MOTENV1 and FP-SNS-FLIGHT1 STM32Cube function packs, STEVAL-MKSBOX1V1 and STSW-WESU1).

The application is available for both Android and iOS platforms and can be found at Android and iOS.

The BlueST-SDK protocol allows easy reception of data over Bluetooth low energy and can be extended to support new data types.

The application and the SDK source codes are released under a BSD license available on GitHub at STBLESensor-Android-src, STBLESensor-iOS-src, STBLESensor-SDK-Aar and STBLESensor-SDK-Ipa.

Table 1. References

<table>
<thead>
<tr>
<th>Order code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STBLESensor-SDK-Ipa</td>
<td>iOS version of BlueST-SDK library that permits easy access to the data exported by a BLE device that implements the BlueST protocol.</td>
</tr>
<tr>
<td>STBLESensor-SDK-Aar</td>
<td>Android version of BlueST-SDK library that permits easy access to the data exported by a BLE device that implements the BlueST protocol.</td>
</tr>
<tr>
<td>STBLESensor-iOS</td>
<td>iOS demo application compatible with BlueST-SDK protocol compatible STM32Cube expansion software.</td>
</tr>
<tr>
<td>STBLESensor-Android</td>
<td>Android demo application compatible with BlueST-SDK protocol compatible STM32Cube expansion software.</td>
</tr>
<tr>
<td>FP-SNS-MOTENV1</td>
<td>Bluetooth low energy and sensors expansion software for STM32Cube.</td>
</tr>
<tr>
<td>FP-SNS-ALLMEMS1</td>
<td>Bluetooth low energy and sensors expansion software for STM32Cube.</td>
</tr>
<tr>
<td>FP-SNS-FLIGHT1</td>
<td>Bluetooth low energy, sensors and NFC tag software expansion for STM32Cube.</td>
</tr>
<tr>
<td>STEVAL-MKSBOX1V1</td>
<td>SensorTile.box wireless multi sensor development kit with user friendly app for IoT and wearable sensor applications</td>
</tr>
</tbody>
</table>
## Acronyms and abbreviations

### Table 2. List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Automatic speech recognition</td>
</tr>
<tr>
<td>BLE</td>
<td>Bluetooth low energy</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma separated values</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics processing unit</td>
</tr>
<tr>
<td>MCU</td>
<td>Microcontroller unit</td>
</tr>
<tr>
<td>MEMS</td>
<td>MicroElectroMechanical systems</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received signal strength indication</td>
</tr>
<tr>
<td>NFC</td>
<td>Near field communication</td>
</tr>
<tr>
<td>SDK</td>
<td>Software development kit</td>
</tr>
</tbody>
</table>
3 Setup

You need an STM32 Nucleo board or/and the relevant expansion boards to set up the chosen expansion software pack.

You can also connect the STEVAL-WESU1, STEVAL-BCNKT01V1, STEVAL-MKSBOX1V1 or the STEVAL-STLKT01V1 evaluation boards.

Android 4.4 or above and iOS 9 or above compatible applications can be found at STBLESensor-Android and STBLESensor-iOS, respectively.
4 Mobile application

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

If you are using the STEVAL-MKSBOX1V1, you can also load or create your custom application.

For further information on this functionality, refer to UM2580, Section 2, freely available at www.st.com.

Figure 1. Main application view

ST BLE Sensor

CONNECT TO A DEVICE

CREATE A NEW APP

ABOUT

ST BLE Sensor
Version 4.4.0
© 2019 STMicroelectronics
4.1 Demos

All the demos share certain features such as the ability to log data and show the output of the serial console. To run a demo:

**Step 1.** Select the **[START LOGGING]** button in the top menu bar or 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.
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
For Android only, you can view or hide the serial console at the page bottom by clicking [Show Serial Console] or [Hide Serial Console].

**Step 2.** 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.

**Note:** 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](image)

**Step 3.** Click on a menu item to show a specific demo view.

4.1.1 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 board is not present the relative data and icon(s) are disabled.

**Step 1.** Click on an image to force the data reading.

Multiple data streams 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).
4.1.2 MEMS Sensor Fusion

To enable this demo the MotionFX library must be present in the system. 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.
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.

**Note:** During the calibration process the proximity data are disabled to avoid a cube flickering while you are rotating the board. If the calibration is not present, it will automatically start when the page is displayed.

To improve the precision of the sensor fusion algorithm, run the calibration process:

**Step 1.** Push the bottom right button to delete existing calibration data and start a new calibration process.

**Step 2.** As the dialog suggests, rotate the board in a figure 8 pattern.
Step 3. Wait for the board to be calibrated (when the bottom right button becomes green).

4.1.2.1 Board movement
To ensure coherent movement between the board and the cube align the starting position:

Step 1. Press the [Reset] button.
Step 2. Keep the board as shown in the image below.
Step 3. Press [OK].
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).

### 4.1.3 Plot data

In this page, you can plot all the data exported by the device.
Step 1. 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.
Figure 11. Selecting the data to plot

Figure 12. Selecting the time scale
4.1.4 SD Logging

This demo is available only when using the FP-SNS-ALLMEMS1 function pack and the STEVAL-STLKT01V1 (SensorTile) board.

This demo allows enabling the data logging into the SensorTile SD card, putting the SensorTile in low power mode during the sampling interval. Since the board is in low power state while the demo is running, data are not sent and all the other demos do not work.

The logging continues working even if the mobile disconnects from the board.

Step 1. Insert the sampling interval.
   The minimum sampling interval is 1 second.

Step 2. Select the data to log.
   The [Start Log] button appears.

Step 3. Press the [Start Log] button.
   If the SD card is correctly detected, the log starts, otherwise an error message is displayed.

4.1.4.1 Disable the SD logging

Step 1. Connect to the board.
Step 2. Select the [SD Logging] demo.
Step 3. Press the [Disable logging] button.
   The logging stops and all the other demo starts working again.

4.1.5 Activity recognition

To enable this demo the MotionAR library must be present in the system.
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 activity.

**Note:** As the algorithm has to collect the data before starting activity recognition, all the images are greyed out for few seconds after the demo starts.

Table 3. Activity recognition values

<table>
<thead>
<tr>
<th>Value</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NO_ACTIVITY</td>
</tr>
<tr>
<td>1</td>
<td>STATIONARY</td>
</tr>
<tr>
<td>2</td>
<td>WALKING</td>
</tr>
<tr>
<td>3</td>
<td>FASTWALKING</td>
</tr>
<tr>
<td>4</td>
<td>JOGGING</td>
</tr>
<tr>
<td>5</td>
<td>BIKING</td>
</tr>
<tr>
<td>6</td>
<td>DRIVING</td>
</tr>
</tbody>
</table>

Figure 14. Activity recognition demo page

4.1.6 Carry position

To enable this demo, the MotionCP library must be present in the system.

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 has to collect the data before starting activity recognition, all the images are greyed out for few seconds after the demo starts.

### Table 4. Carry position values

<table>
<thead>
<tr>
<th>Value</th>
<th>Carry position</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>1</td>
<td>ON_DESK</td>
</tr>
<tr>
<td>2</td>
<td>IN_HAND</td>
</tr>
<tr>
<td>3</td>
<td>NEAR_HEAD</td>
</tr>
<tr>
<td>4</td>
<td>SHIRT_POCKET</td>
</tr>
<tr>
<td>5</td>
<td>TROUSERS_POCKET</td>
</tr>
<tr>
<td>6</td>
<td>ARM_SWING</td>
</tr>
</tbody>
</table>

4.1.7 Gesture recognition

To enable this demo, the MotionGR library must be present in the system.

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 the device up
- 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 occurs, the icon becomes grey again.
Table 5. Gesture recognition values

<table>
<thead>
<tr>
<th>Value</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>1</td>
<td>PICK_UP</td>
</tr>
<tr>
<td>2</td>
<td>GLANCE</td>
</tr>
<tr>
<td>3</td>
<td>WAKE_UP</td>
</tr>
</tbody>
</table>

Figure 16. Mems gesture demo page

4.1.8 Pedometer
To enable this demo, the MotionPM library must be present in the system.
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.
4.1.9 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/LSM6DSL 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**: provides the current chip orientation. The related image will show the chip with the dot marker oriented as the real one.
- **Free fall**: fired when the board hits the bottom after a free fall.
- **Single tap**: fired when the user taps on the board.
- **Double tap**: fired when the user performs a double tap on the board in a rapid sequence.
- **Wake-up**: fired when the user moves the board.
- **Pedometer**: fired when a step is detected. The chip is also able to count the number of steps performed after the event detection starts.

**Note**: The chip does not signal the first six steps identified by the pedometer.

- **Tilt**: 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.
- **Multiple**: the orientation, pedometer, single tap, free fall, double tap and tilt events are enabled at the same time.
### Table 6. Acceleration event values

<table>
<thead>
<tr>
<th>Value</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x000</td>
<td>NO_EVENT</td>
</tr>
<tr>
<td>0x001</td>
<td>ORIENTATION_TOP_RIGHT</td>
</tr>
<tr>
<td>0x002</td>
<td>ORIENTATION_BOTTOM_RIGHT</td>
</tr>
<tr>
<td>0x003</td>
<td>ORIENTATION_BOTTOM_LEFT</td>
</tr>
<tr>
<td>0x004</td>
<td>ORIENTATION_TOP_LEFT</td>
</tr>
<tr>
<td>0x005</td>
<td>ORIENTATION_UP</td>
</tr>
<tr>
<td>0x006</td>
<td>ORIENTATION_DOWN</td>
</tr>
<tr>
<td>0x008</td>
<td>TILT</td>
</tr>
<tr>
<td>0x010</td>
<td>FREE_FALL</td>
</tr>
<tr>
<td>0x020</td>
<td>SINGLE_TAP</td>
</tr>
<tr>
<td>0x040</td>
<td>DOUBLE_TAP</td>
</tr>
<tr>
<td>0x080</td>
<td>WAKE_UP</td>
</tr>
<tr>
<td>0x100</td>
<td>PEDOMETER</td>
</tr>
</tbody>
</table>

![Figure 18. Orientation event](image)
Figure 19. Free fall event

Figure 20. Single Tap event
**Figure 21. Double Tap event**

Detected event: DOUBLE_TAP

**Figure 22. Wake Up event**

Detected event: WAKE_UP
Figure 23. Pedometer event

Figure 24. Tilt event
4.1.10 LED switch
This panel can detect and change the status of the user LED on the board.

**Step 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 25. Switch demo page](image)

4.1.11 BlueVoice
If the BlueVoice library is enabled, the following page is also available.
Supported functions are:

- Playback of the audio stream received from the connected device.
- Beam forming can be enabled to use a virtual directional microphone when multiple microphones are available.
- Showing the audio wave plot.
- Possibility of audio recording in a wav file, instead of logging as a CSV file.
- Audio playback begins as soon as the page is displayed.

**Step 1.** Adjust the volume level using the slider or mute by clicking on the speaker icon.

**4.1.12 Speech to text**

If the BlueVoice library is enabled, the following page is also available. This page is dedicated to the cloud speech to text services. Currently four services are supported:

- Google for English, Italian, French, Spanish, German and Portuguese
- iFlyTek for Chinese
- IBM Watson for American and UK English accent
- Generic websocket to stream the raw audio data (16 bit little endian, with a sampling rate of 8 kHz) to a private websocket server
Step 1. Click on the [Select] button to select the speech to text provider and the language to use.

Step 2. If the selected provider needs it, press the key button to set service login credential.
See Section 6.3 Speech to text activation and Section 5 Google speech ASR Key generation for the instruction needed to obtain the service login.

Step 3. To start the audio streaming press the microphone icon or double tap the board if it supports acceleration events.
As the Google service does not support a continuous stream, you have to keep pressing the icon to register a message which will be sent to the cloud when you release the button.
The Google service returns a confidence result, that is: the app shows recognized text only if confidence is bigger than 75%, otherwise an error message appears.
When the button turns green, the connection with the service is established and you can start speaking.

Step 4. To stop the audio streaming press again the microphone button or double tap the board.

4.1.13 Beamforming
To enable this demo the AcusticBL library and different microphones must be present in the system.
The beamforming algorithm combines the audio from the microphones to create a virtual directional microphone, whose direction can be changed using the button around the board image (see the figure below).
The X-NUCLEO-CCA02M1 expansion board has two built-in microphones that allow choosing between two directions. The STEVAL-BCNKTO1V1 has four microphones and eight available directions.
At the bottom of the page, a plot with the audio intensity is available; speaking in a perpendicular direction with respect to the one selected makes the sound wave low since it is filtered out by the algorithm.
The transmitted audio can be recorded in a wav file to further analyze the algorithm output.
4.1.14 Source localization

To enable this demo the AcusticSL library and different microphones must be present in the system. This page enables the Source Localization algorithm which finds the sound source direction. The switch changes the algorithm sensitivity: if set to high, the algorithm detects sounds like a human voice with a normal tone; if set to low, it detects just loud sounds.
4.1.15 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 on-board magnetometer. The calibration process starts automatically if the board is not calibrated yet.
4.1.16 Motion intensity

To enable this demo the MotionID library must be present in the system. 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.
4.1.17 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).
The RSSI value is updated every second.

4.1.18 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 Generic MQTT, Azure IOT Hub, AWS IoT, and 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.

The cloud update interval can be changed before opening the connection through the [Update interval] menu.

By default, data are sent to the cloud only if more than 5 seconds elapsed from the last update.
4.1.18.1 **IBM Watson Quickstart**

Figure 33. **Data needed for the Quickstart service**
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 webpage. The Device ID must be unique and the default value is the device name and the last 3 bytes of the MAC address.

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

• click on the [View Cloud Data] button and open the webpage from the mobile phone;
• visit the page https://quickstart.internetofthings.ibmcloud.com.

**Note:** In the webpage you have to accept the terms of service, insert the [Device ID] and press the [Go] button to show the plot.
4.1.18.2 IBM Watson registered account

Step 1. Select [IBM Watson] option from the menu to use the related platform. Before connection starts, the following fields must be filled in mandatorily:

- organization ID: the name of the organization 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

Step 2. Click on the [View Cloud Data] button to view cloud data and open the webpage from the mobile phone or visit https://[organization].internetofthings.ibmcloud.com/dashboard.

You can also use a demo ("play") account (refer to Section 6 IBM Watson IoT for further details).

4.1.18.3 Azure IoT Hub

Step 1. Select [Azure IoT - ST Web Dashboard] from the menu to use the ST account inside the Azure IoT platform.

Step 2. Press the [Register Device] button to register a device or retrieve the device information.
Figure 36. Azure IoT demo: register the device

Step 3. Click at the bottom to open a connection.
Step 4. Select the data to upload.
Step 5. Click on the [View Cloud Data] button to view the uploaded data.
Figure 37. Azure IoT demo: view cloud data

Step 6. Open the webpage from the mobile phone or visit https://stm32ode.azurewebsites.net
Step 7. Insert the board MAC address to log in.

4.1.18.3.1 Firmware upgrade

To upgrade the firmware, follow the steps below.

Step 1. Click the [Device management] button.
Step 2. Select the right firmware for your platform and press the [Force firmware upgrade] button.
Step 3. Upload the firmware clicking the [Choose file] button.
A notification appears on you mobile.

**Step 4.** Press the [Start firmware upgrade] button to download and launch the flashing procedure.

### 4.1.18.4 **AWS IoT**

To use this demo, after selecting [AWS IoT] from the option menu, you have to create an account on AWS and register a new thing.

To register a new thing, refer to the Amazon documentation:
1. **Register a new Thing**: the thing name has to be inserted in the app under the [Client Id] field.
2. **Create the certificate**: download the certificate and the private key to load to the app.
3. **Create the permission policy to be used by the thing**: for this specific, the [iot:Connect] and [iot:Publish].

The resulting policy should be similar to:

```json
{
  "Version": "2012-10-17",
  "Statement": [  
    {  
      "Effect": "Allow",
      "Action": "iot:Connect",
    },  
    {  
      "Effect": "Allow",
      "Action": "iot:Publish",
    }
  ]
}
```

**Note:** Change the <CHANGE_ME> string with your thing name and replace the <region> and <id> using your AWS region and your Used Id.

4. **Attach the policy to the certificate**
5. **Attach the certificate to the thing**

The AWS endpoint needed by the app is located at: [Manage]|[Things]|[thing name]|[Interact]|[Https]. You have to copy the address in the form: `<code>.iot.<Aws region>.amazonaws.com`
Once all the data are inserted and the connection is established, you can start sending data selecting the desired feature. The data are uploaded in the topic `<Client Id>/<Data Name>` using the JSON format `{<field Name>:<field Value>, ...}`. You can visualize the data from the AWS console selecting the [Test menu] and subscribing to the topic `#` (`#` is a wildcard that means all topics).

### 4.1.18.5 Generic MQTT

To use a private MQTT server select [Generic MQTT] from the option menu. To enable the connection you have to insert:

- the server address, with the transmission protocol used which can be tcp for the insecure connection or ssh for the secure connection;
- the port used by the MQTT server;
- if needed, the user and password to connect to the server;
- the client Id used to identify the peripheral in the server.

The data are uploaded in the topic `<Client Id>/<Data Name>/<Field Name>` encoding the values as strings.

### 4.2 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 expansion board display

**Note:** 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.

### 4.2.1 NFC

Android platforms have settings 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 STBLESensor 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 not using the NFC tag functionality).

**Note:** The messages do not always appear in the foreground; you may have to access them via the Notification Center.
4.3 Debug console

Cicking [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 echoes 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.

Note: 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.
4.4 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**

- **Export Path**
  Select the path where save/dump the log

- **Log storage**
  Select where store the log

- **Clear Log**
  Remove previous logs

---

**Figure 42. Path to save the log data**

- **Export Path**
  
  `/sdcard/`

  - Cancel
  - OK
4.5 License Manager

For the functions requiring a valid license, requests must be made for each STM32 Nucleo board where the firmware will be flashed. STBLESensor 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 functionally, select the License Manager from the menu in the demo view.
**Figure 44. Main License Manager view**

Available licenses are:
- osxMotionFX
  - Mem sensor fusion
- osxMotionAR
  - Activity recognition
- osxMotionCP
  - Carry position
- osxMotionGR
  - Gesture recognition
- osxMotionPM
  - Pedometer

**Figure 45. License Manager menu**

Available licenses:
- osxMotionFX
  - Mem sensor fusion
- osxMotionAR
  - Activity recognition
- osxMotionCP
  - Carry position
- osxMotionGR
  - Gesture recognition
- osxMotionPM
  - Pedometer

Options:
- Clear app license DB
- Clear board licenses
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.

4.5.1 Requesting a license

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

Step 1. Click [Request] to request a new license.

Step 2. Read the license agreement and then click the [Agree] button if you wish to proceed.

Step 3. A new view appears where you need to fill in the relevant information.

Note: Avoid using non-Latin characters.
User data is saved by the application so you do not have to enter it again for subsequent licenses.

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

Figure 46. Example of license to agree with
4.5.2 Uploading the license

Step 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.
Figure 48. Screen where the license code must be pasted

![License Code Screen](image)

Figure 49. Text to copy and paste in BlueMS

```c
#define MOTIONFx_LICENSE \
#define MOTIONFx_LICENSE \
uint32_t _mfx_license[3][4] = \n{ \n    { 0xAB10C349, 0xEB830C9B, 0x4494359B, 0x987359ED }, \n    { 0x9C58E206, 0x780E791B, 0xFA632A5, 0xE5845B34F }, \n    { 0x5225F59B, 0x881A5D6F, 0x96C274F8, 0xA278257 } \n};
#endif // _MOTIONFx_LICENSE
```

Thanks for your interest in the ST MOTIONFx_V1.40 software library.
Please find here below your personal activation license (node-locked).

To use the above activation license just copy the text between the lines
and paste it into the local “license.h” file on your PC. The file is located
in the “Middlewares\ST\Library_Name” folder.

Enjoy developing with ST OpenSoftwareX and have
great ideas turn into life.Augmented.

Best regards.
A dialog informs the user whether the license has been correctly uploaded and accepted by the system.

Figure 50. License loaded confirmation message

Step 2. 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.

4.6 Firmware upgrade

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

Step 1. Click on [Firmware upgrade] from the menu in the demo view to start the firmware upgrade procedure.
The main view shows information regarding the current firmware version.
Step 2. Click on the [Upload] button and select the binary file to upload. The upload process begins immediately.

*Note:* 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
Once the new firmware is correctly uploaded to the board, the board restarts.

**Step 3.** Restart the STBLESensor application.

**Figure 54. Uploading successfully completed**

![Image of firmware upgrade successful completion]

**BLUEMICROSYSTEM1**

- Version: 3.0.1
- Mcu Type: F401

Flash Completed in: 116.88s
The board is resetting
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. In this case:

a. reset the board
b. wait for the application to acknowledge the disconnection
c. 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.
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:

**Step 1.** Login with your own Google account.

**Step 2.** Subscribe to Chromium-dev at https://groups.google.com/a/chromium.org/forum/?fromgroups#!forum/chromium-dev.

**Step 3.** Write “Chromium-dev” in the search box, and select the appropriate group.

![Google Chromium-dev: search group](image)

**Step 4.** Click on “Join group to post” button

![Google Chromium-dev: join group to post](image)

**Step 5.** Click on “Join this group” button to join the Chromium-dev group.

![Google Chromium-dev: join the group](image)

**Step 6.** Go to https://console.developers.google.com/project

**Step 7.** Click on “Create a project…”
Step 8. Choose the Project name.
Step 9. Click on “Create” button.

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

Step 11. Write “Speech API” in the search box, and select correct result.

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

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

Figure 59. Google Chromium-dev: create project

Figure 60. Google Developers Console: new project

Figure 61. Google Developers Console: ASRProject

Figure 62. Google Developers Console: select API

Figure 63. Google API Manager: enable API
**Figure 64. Google API Manager: create API key**

Step 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**
6 IBM Watson IoT

6.1 Register a device

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

To register the device:

Step 2. Click on [Add Device].

![Figure 66. Watson IoT platform: Add Device]

Step 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]

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

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

Step 6. Confirm all the data.

6.2 Display data

When using the STBLESensor app, you should enter the device ID/type, the authentication token, and “play” as the organization name in the proper fields.
To display the board:

**Step 1.** Visit https://play.internetofthings.ibmcloud.com/dashboard/.

**Step 2.** Open or create a new board configuration.

**Step 3.** Add a new card.

![Figure 68. Watson IoT platform: Add New Card](image)

**Step 4.** Select the linear plot card.

**Step 5.** Search the related device ID and select it.

**Step 6.** Add a new dataset that will be displayed in the plot.
If the board is already sending data, you can choose the event and property from the menu.

![Figure 69. Watson IoT platform: add a new dataset](image)

**Step 7.** Press [Next].
You can see data coming from the node in real-time. The plot type, size, name and color can also be changed.

### 6.3 Speech to text activation

To use the IBM Watson speech to text service, user and authentication key are necessary.
To get the login data:

**Step 1.** Visit https://console.bluemix.net/va.
Step 2. Login or create a free account.
Step 3. Once you are logged in, press the [Create resource] blue button.
Step 4. Search for [Speech to Text] and select:

![Figure 70. Speech to text card](image)

Step 5. Click on [Create] button at the bottom-right corner of the window.
Step 6. Once you have created the resource, click on [Show].
Step 7. Copy the obtained credentials.

![Figure 71. Credentials to use in the STBLESensor app](image)
## Revision history

Table 7. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>22-May-2019</td>
<td>2</td>
<td>Updated Introduction and Section 2 Acronyms and abbreviations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Section 1 Overview.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added STEVAL-MKSBOX1V1 compatibility information.</td>
</tr>
</tbody>
</table>
Contents

1 Overview .......................................................................... 2
2 Acronyms and abbreviations ...................................................... 3
3 Setup. .............................................................................. 4
4 Mobile application ................................................................. 5
   4.1 Demos ....................................................................... 6
      4.1.1 Environmental ........................................................... 8
      4.1.2 MEMS Sensor Fusion ..................................................... 9
      4.1.3 Plot data ..................................................................... 12
      4.1.4 SD Logging .................................................................. 14
      4.1.5 Activity recognition ...................................................... 15
      4.1.6 Carry position .......................................................... 16
      4.1.7 Gesture recognition...................................................... 17
      4.1.8 Pedometer ............................................................ 18
      4.1.9 Acceleration events. .................................................... 19
      4.1.10 LED switch ............................................................ 23
      4.1.11 BlueVoice ............................................................. 24
      4.1.12 Speech to text.......................................................... 25
      4.1.13 Beamforming........................................................... 26
      4.1.14 Source localization ...................................................... 27
      4.1.15 Compass.............................................................. 28
      4.1.16 Motion intensity ......................................................... 29
      4.1.17 Rssi and battery ........................................................ 30
      4.1.18 Cloud ................................................................ 31
   4.2 Secure pairing ................................................................ 38
      4.2.1 NFC ................................................................. 38
   4.3 Debug console. ............................................................. 39
   4.4 Log settings .................................................................. 40
   4.5 License Manager.............................................................. 42
      4.5.1 Requesting a license. .................................................... 44
      4.5.2 Uploading the license .................................................... 45
4.6  Firmware upgrade ............................................................................................................................. 47

5  Google speech ASR Key generation ..................................................................................................... 52

6  IBM Watson IoT ................................................................................................................................. 55

   6.1  Register a device ........................................................................................................................... 55
   6.2  Display data ................................................................................................................................. 55
   6.3  Speech to text activation ............................................................................................................... 56

Revision history ........................................................................................................................................ 58
List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Main application view</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Android application scanning results</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>The application asks to send the acquired log by e-mail</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>The serial console is active and shows the data sent by the device</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Example demo menu</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>Environmental page</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>MEMS Sensor Fusion page</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>MEMS Sensor fusion calibration dialog</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>MEMS Sensor fusion reset board position</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>Plot Data page</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>Selecting the data to plot</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>Selecting the time scale</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>SD logging demo page</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>Activity recognition demo page</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>Carry position demo page</td>
<td>17</td>
</tr>
<tr>
<td>16</td>
<td>Mems gesture demo page</td>
<td>18</td>
</tr>
<tr>
<td>17</td>
<td>Pedometer demo page</td>
<td>19</td>
</tr>
<tr>
<td>18</td>
<td>Orientation event</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>Free fall event</td>
<td>21</td>
</tr>
<tr>
<td>20</td>
<td>Single Tap event</td>
<td>21</td>
</tr>
<tr>
<td>21</td>
<td>Double Tap event</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>Wake Up event</td>
<td>22</td>
</tr>
<tr>
<td>23</td>
<td>Pedometer event</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>Tilt event</td>
<td>23</td>
</tr>
<tr>
<td>25</td>
<td>Switch demo page</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>BlueVoice Demo page</td>
<td>25</td>
</tr>
<tr>
<td>27</td>
<td>Speech to text demo page</td>
<td>26</td>
</tr>
<tr>
<td>28</td>
<td>Beamforming demo page</td>
<td>27</td>
</tr>
<tr>
<td>29</td>
<td>Source localization demo page</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>Compass demo page</td>
<td>29</td>
</tr>
<tr>
<td>31</td>
<td>Motion intensity demo page</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>Board status demo page</td>
<td>31</td>
</tr>
<tr>
<td>33</td>
<td>Data needed for the Quickstart service</td>
<td>32</td>
</tr>
<tr>
<td>34</td>
<td>List of available features after the connection</td>
<td>33</td>
</tr>
<tr>
<td>35</td>
<td>IBM Watson Quickstart webpage</td>
<td>34</td>
</tr>
<tr>
<td>36</td>
<td>Azure IoT demo: register the device</td>
<td>35</td>
</tr>
<tr>
<td>37</td>
<td>Azure IoT demo: view cloud data</td>
<td>36</td>
</tr>
<tr>
<td>38</td>
<td>Azure IoT demo: device management page</td>
<td>37</td>
</tr>
<tr>
<td>39</td>
<td>PIN prompt dialog</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>Debug serial console</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>Settings screen</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>Path to save the log data</td>
<td>41</td>
</tr>
<tr>
<td>43</td>
<td>Dialog to select the log format</td>
<td>42</td>
</tr>
<tr>
<td>44</td>
<td>Main License Manager view</td>
<td>43</td>
</tr>
<tr>
<td>45</td>
<td>License Manager menu</td>
<td>43</td>
</tr>
<tr>
<td>46</td>
<td>Example of license to agree with</td>
<td>44</td>
</tr>
<tr>
<td>47</td>
<td>Fields to compile to request the license</td>
<td>45</td>
</tr>
<tr>
<td>48</td>
<td>Screen where the license code must be pasted</td>
<td>46</td>
</tr>
<tr>
<td>49</td>
<td>Text to copy and paste in BlueMS</td>
<td>46</td>
</tr>
<tr>
<td>50</td>
<td>License loaded confirmation message</td>
<td>47</td>
</tr>
<tr>
<td>51</td>
<td>Firmware upgrade main view</td>
<td>48</td>
</tr>
<tr>
<td>52</td>
<td>Formatting the Flash memory</td>
<td>49</td>
</tr>
</tbody>
</table>
Figure 53. Uploading the new firmware .......................................................... 49
Figure 54. Uploading successfully completed ...................................................... 50
Figure 55. Error during the uploading ........................................................... 51
Figure 56. Google Chromium-dev: search group ................................................... 52
Figure 57. Google Chromium-dev: join group to post ................................................ 52
Figure 58. Google Chromium-dev: join the group ................................................... 52
Figure 59. Google Chromium-dev: create project ................................................... 52
Figure 60. Google Developers Console: new project ................................................. 53
Figure 61. Google Developers Console: ASRProject ................................................ 53
Figure 62. Google Developers Console: select API .................................................. 53
Figure 63. Google API Manager: enable API ...................................................... 53
Figure 64. Google API Manager: create API key ................................................... 54
Figure 65. Google API Manager: Android API key ................................................... 54
Figure 66. Watson IoT platform: Add Device ...................................................... 55
Figure 67. Watson IoT platform: Choose Device Type ................................................ 55
Figure 68. Watson IoT platform: Add New Card .................................................... 56
Figure 69. Watson IoT platform: add a new dataset ................................................. 56
Figure 70. Speech to text card ................................................................ 57
Figure 71. Credentials to use in the STBLESensor app ................................................. 57
# List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>Table 2</td>
<td>List of acronyms</td>
<td>3</td>
</tr>
<tr>
<td>Table 3</td>
<td>Activity recognition values</td>
<td>16</td>
</tr>
<tr>
<td>Table 4</td>
<td>Carry position values</td>
<td>17</td>
</tr>
<tr>
<td>Table 5</td>
<td>Gesture recognition values</td>
<td>18</td>
</tr>
<tr>
<td>Table 6</td>
<td>Acceleration event values</td>
<td>20</td>
</tr>
<tr>
<td>Table 7</td>
<td>Document revision history</td>
<td>58</td>
</tr>
</tbody>
</table>