Getting started with the STM32Cube function pack for STEVAL-STWINKT1B evaluation kit plus STEVAL-STWINWFV1 Wi-Fi adapter board for predictive maintenance application based on artificial intelligence (AI)

Introduction

FP-AI-PREDMNT2 is an STM32Cube function pack that programs the STWIN as an IoT Edge node, connected to the cloud, able to acquire sensor data, process them and send the results to the DSH-PREDMNT cloud dashboard. It includes dedicated algorithms for advanced time and frequency domain signal processing and analysis of 3D digital accelerometers with flat bandwidth up to 6 kHz.

The function pack helps to jump-start the implementation and development of condition monitoring applications designed with the NanoEdge™ AI Studio solution, thus easily enabling an AI-based predictive maintenance solution (the NanoEdge™ AI library generation is out of the scope of this function pack and must be generated using NanoEdge™ AI Studio).

The package includes pressure, relative humidity and temperature sensor monitoring, as well as audio algorithms to check acoustic emission (AE), up to 20 kHz, and ultrasound emission analysis up to 80 kHz.

Using the STBLESensor app you can set up Wi-Fi credentials and exchange cloud certificates to enable the connection to the dedicated DSH-PREDMNT web-based dashboard. The dashboard allows monitoring and logging the algorithm output, sensor data and equipment status.

The FP-AI-PREDMNT2, together with the suggested combination of STM32 and ST devices, can be used to develop specific industrial predictive maintenance applications for early detection of warning signs of potential failure.

The software runs on the STM32 microcontroller and includes all the necessary drivers for the STEVAL-STWINKT1B evaluation kit.

FP-AI-PREDMNT2 firmware is based on application-level modules (Sensor Manager, Digital Processing Units, etc.) that you can reuse and easily extend to build a customized application.

These application modules adopt state-of-the-art design patterns and natively support low-power modes. To enable this solution, the function pack has been built on top of eLooM, an embedded Light object-oriented fraMework for STM32 applications specifically designed for embedded low-power applications powered by STM32.

Related links

Visit the STM32Cube ecosystem web page on www.st.com for further information

1 FP-AI-PREDMNT2 software expansion for STM32Cube

1.1 Overview

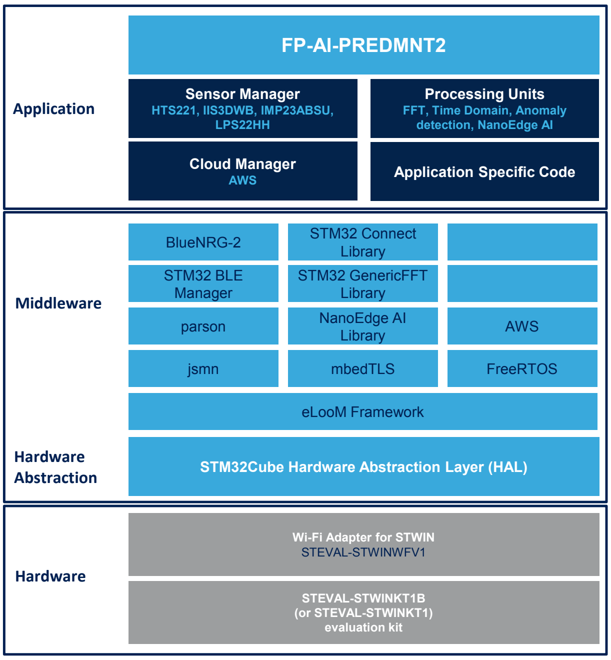
The FP-AI-PREDMNT2 key features are:

- Complete firmware to develop a sensor node for predictive maintenance applications, featuring analog microphone, environmental and motion sensors, and performing real-time monitoring of parameters and equipment status via Wi-Fi connectivity
- Compatible with NanoEdge™ AI Studio solution, to enable AI-based solution
- Generic FFT library middleware to enable frequency domain analysis for any kind of sensor through Fast Fourier Transform (with programmable size, overlapping and windowing)
- Motion TD library middleware for vibration analysis in time domain (speed RMS and acceleration peak)
- Configurable alarm and warning thresholds for key parameters
- Compatible with STBLESensor application for Android/iOS, to perform Wi-Fi configuration and secure certificate provisioning
- Compatible with DSH-PREDMNT web-based predictive maintenance dashboard for monitoring sensor data and device status
- Easy portability across different MCU families, thanks to STM32Cube
- Firmware modular example based on eLooM (embedded Light object-oriented fraMework for STM32) to enable code re-usability at application level
- Free, user-friendly license terms

1.2 Architecture

The FP-AI-PREDMNT2 software is designed for the STEVAL-STWINKT1B development kit with the STEVAL-STWINWFV1 Wi-Fi board.

Figure 1. FP-AI-PREDMNT2 software architecture



The FP-AI-PREDMNT2, compliant with STM32Cube architecture, is structured into a set of layers of increasing abstraction.

The hardware abstraction layer (HAL) interfaces with the hardware and provides the low level drivers and the hardware interfacing methods to interact with the upper layers (application, libraries and stacks). It provides APIs for the communication peripherals (I²C, SPI, UART, etc.) for initialization and configuration, data transfer and communication errors.

There are two types of HAL driver APIs:

- generic APIs which provide common and generic functions to the entire STM32 series
- extension APIs which provide specific, customized functions for a particular family or a specific part number

The package extends STM32Cube by providing a board support package (BSP) which deals with board specific peripherals and functions (LED, user button, etc.).

The BSP structure follows the hardware structure, including a component management layer as well as the specific layers of the boards used.

On top of such features, inherited from STM32Cube, FP-AI-PREDMNT2 adds code re-usability at application level.

FP-AI-PREDMNT2 is an eLooM-based application-level firmware. It is based on several firmware modules that interface and offer data to other application modules according to well defined design patterns and specific APIs.

Each firmware module (i.e.: Sensor Manager, Digital Processing Unit) is packed into a folder. They are independent from each other and they can be added to your custom firmware application by just dragging and dropping the needed folder.

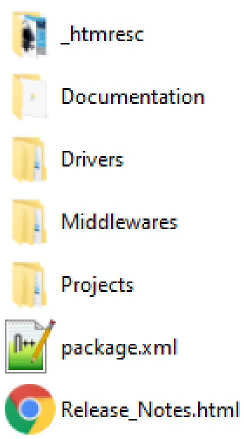
Each firmware module implements or extends services, classes and objects made available by the eLooM framework. More specifically, here you can find:

- drivers - objects that implement the base interface for any low-level subsystem that can be used into the firmware module (i.e.: I2C, DFSDM)
- events - objects that handle information about something that happened in the system at a given moment. These files implement the event and source/listener design patterns
- services - additional utilities for the firmware module

1.3 Folders

The FP-AI-PREDMNT2 firmware package folder structure follows the layer-based approach of the STM32Cube architecture.

Figure 2. FP-AI-PREDMNT2 package folder structure



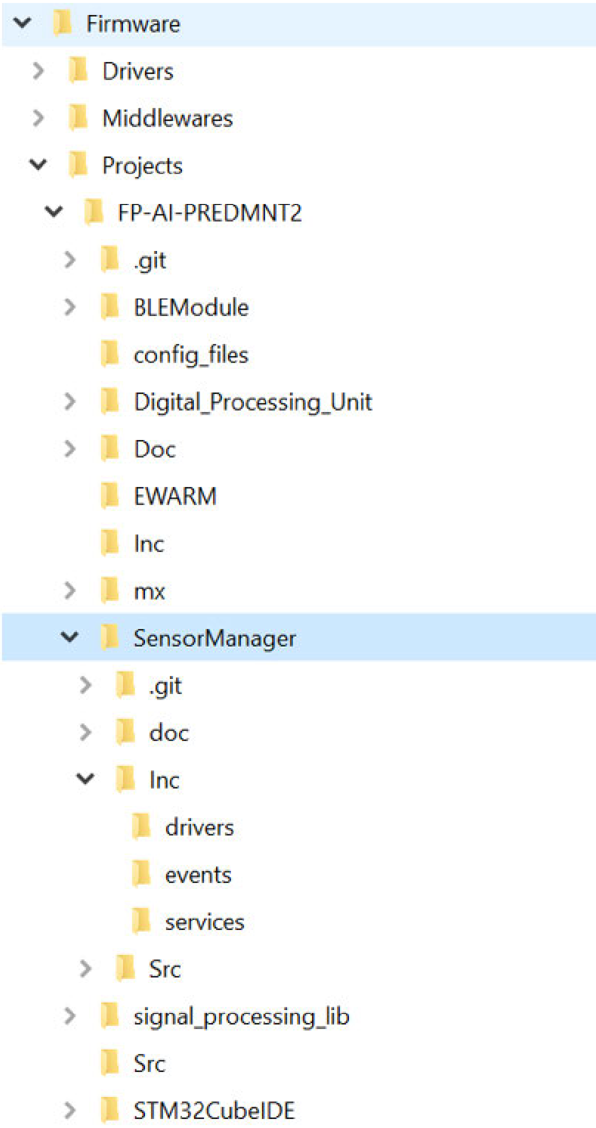
The folders included in the software package are:

- **Documentation** contains a compiled HTML file generated from the source code which details the software components and APIs.
- **Drivers** contains the HAL drivers and the board-specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS vendor-independent hardware abstraction layer for ARM Cortex-M processor series.

- **Middlewares** contains libraries and protocols for BlueNRG-2 Bluetooth low energy, BLEManager library, eLooM (embedded Light object-oriented fraMework) for STM32, GenericFFT and MotionTD software libraries, USB Device Library, STM32 Connect Library which provides API to access network services on STM32 devices, NanoEdgeAI library, Parson library, jsmn library, AWS library and FreeRTOS real-time operating system.
- **Projects** contains sample application used to transmit the output of the sensor data, the analysis results of the motion processing and ultrasound analysis over Wi-Fi.

  This application is provided for the STEVAL-STWINKT1B evaluation kit with the STEVAL-STWINWFV1 Wi-Fi board with three development environments (IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM), and STM32CubeIDE).

Figure 3. FP-AI-PREDMNT2 module sub-folders



1.4 APIs

Detailed technical information with full user API function and parameter description are in a compiled HTML file in the “Documentation” folder and in the "Doc" folder of each firmware module and eLooM middleware.

2 Sample application description

The predictive maintenance cloud application is a software example for STWIN (STEVAL-STWINKT1B plus STEVAL-STWINWFV1) that collects sensor data and performs spectral analysis and AI-based condition monitoring for the IIS3DWB accelerometer (up to 6 kHz) and the IMP23ABSU analog microphone (up to 80 kHz) signals.

The example is provided in the “Projects” directory. Ready to be built projects are available for multiple IDEs.

The result of spectral analysis and raw environmental data is sent to the DSH-PREDMNT dashboard exploiting AWS (Amazon Web Service) SDK and MQTT protocol.

You can set the predictive maintenance cloud application parameters through the dashboard.

The application mainly:

- performs a secure connection to DSH-PREDMNT
- computes the spectrum of audio signal from analog microphone in ultrasound bandwidth up to 80 kHz
- analyzes vibrations by computing the spectrum of vibrometer data up to 6 KHz
- gets the environmental sensor data
- receives configuration parameters from the dashboard
- sets up parameters and starts the training and detection phases of a NanoEdge AI library (from the dashboard)
- provides firmware based on eLooM application-level modules

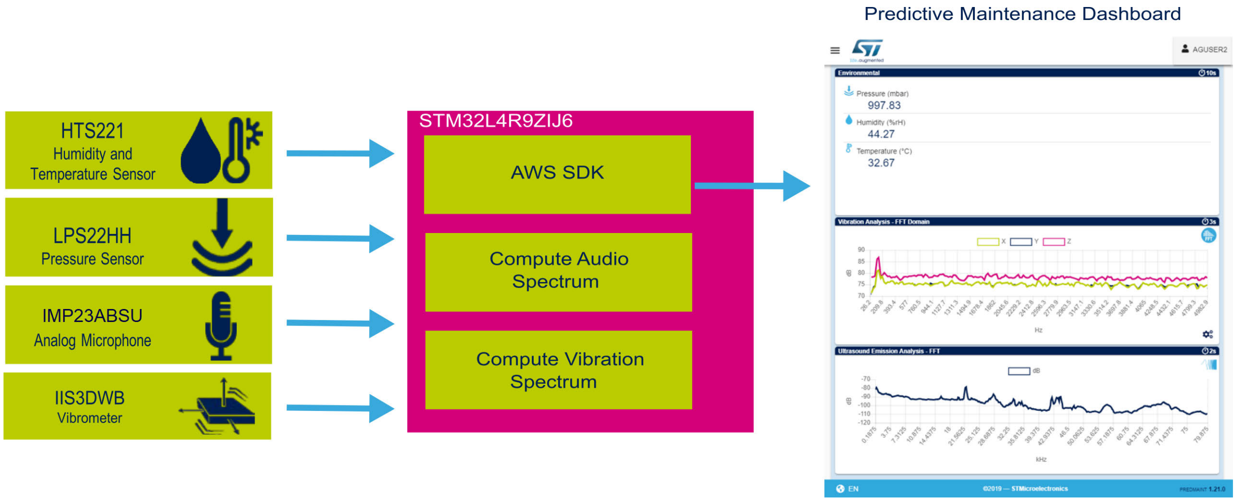
2.1 Functional blocks

The predictive maintenance cloud application connects to the DSH-PREDMNT dashboard exploiting the underlying AWS IoT Core API and defines the following AWS MQTT topics:

- environmental data topic:
  - temperature and humidity (HTS221)
  - pressure (LPS22HH)
- ultrasound analysis topic:
  - audio spectral analysis in ultrasound bandwidth exploiting IMP23ABSU analog microphone
- vibration analysis topic:
  - vibration analysis up to 5 kHz of bandwidth exploiting IIS3DWB MEMS vibrometer

Data are then collected and processed in independent tasks.

Figure 4. Predictive maintenance cloud application functional blocks



2.2 Setup of the NanoEdge™ AI machine learning library

FP-AI-PREDMNT2 includes a pre-integrated stub of a Nanoedge AI library, which can be easily replaced by the user with a real customized library generated with NanoEdge™ AI Studio. This stub simulates the NanoEdge™ AI-related functionalities, such as running, learning, and detection phases on the edge.

The following section shows how to generate a NanoEdge™ AI library for anomaly detection using NanoEdge™ AI Studio and FP-AI-PREDMNT2, and how to set up an application for condition monitoring.

2.2.1 Generating a condition monitoring library

2.2.1.1 Data logging for normal and abnormal conditions

The data collection functionality is out of the scope of this function pack. However, to generate a custom NanoEdge™ AI library you must feed the NanoEdge™ AI Studio with a proper training dataset, composed of one dataset for normal condition and one for abnormal condition.

Note: *During training, expose the AI model/algorithm to all the normal and abnormal cases available. For further information, refer to the detailed documentation of the NanoEdge™ AI Studio.*

To prepare this training dataset, you can use HSDatalog firmware.

Flash your STWIN board with HSDatalog firmware, make a log, prepare data for library generation, and flash again the STWIN board as described in the following paragraphs.

Note: *For details on how to use all the features of HSDatalog firmware, refer to the user manual of FP-SNS-DATALOG1.*

2.2.1.2 Data preparation for library generation with NanoEdge™ AI Studio

The data logged through the datalogger is in binary format and is not user readable nor compliant with the NanoEdge™ AI Studio format as it is. To convert this data, FP-SNS-DATALOG1 provides Python™ utility scripts in `/FP-SNS-DATALOG1/Utilities/Python_SDK/`.

Just run the `hsdatalog_to_nanoedge.py` Python™ script in your preferred Python™ environment or from cli (for usage help, type python hsdatalog_to_nanoedge.py -h) to prepare segments for a given data acquisition and generate the related .csv files.

The conversion should end with the creation of two files, one for the normal conditions (`normal.csv`) and one for the anomalous condition (`abnormal.csv`).

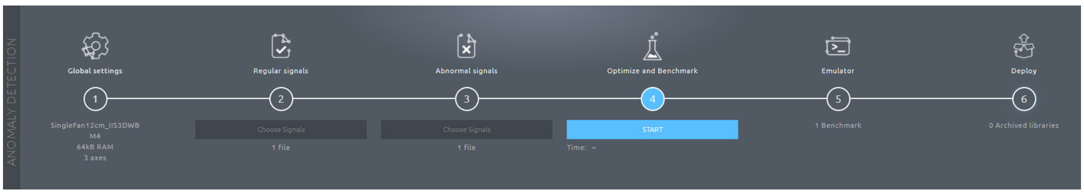
You are free to rename these files as you prefer.

Note: *Fully detailed information on Python™ SDK features are available in paragraph 2.6.2 of the user manual of FP-SNS-DATALOG1.*

2.2.1.3 Library generation using NanoEdge™ AI Studio

The process of generating the libraries with NanoEdge™ AI Studio, using normal and abnormal segment files, consists of six steps.

Figure 5. Library generation steps



Step 1. Choose the target platform or a microcontroller type: STWIN Cortex-M4.

Step 2. Define the maximum amount of RAM to allocate for the library.
Usually a few Kbytes are enough but it depends on the data frame length used in the process of data preparation. 32 Kbytes is a good starting point.

Step 3. Select the sensor type: 3-axis accelerometer.

Step 4. Provide the sample contextual data for normal segments to adjust and gauge the performance of the chosen model (`normal.csv`).

Step 5. Provide the sample contextual data for abnormal segments to adjust and gauge the performance of the chosen model (`abnormal.csv`).

Step 6. Benchmark available models and choose the one that complies with the requirements.

Step 7. Validate the model to learn and test through the provided emulator, which emulates the behavior of the library on the edge.

Step 8. Compile and download the libraries.

Important: *Check the `-mfloat-abi` flag to use the libraries with hardware FPU and `-fshort-wcharand` to use the library with µKeil. Leave the other flags in their default state.*

Note: *NanoEdge™AI libraries can be freely generated for evaluation purposes for STEVAL-STWINKT1B Cortex-M4, NUCLEO-F401RE Cortex-M4, NUCLEO-L432KC Cortex-M4, and STM32L562E-DK Cortex-M33.*

2.2.2 Installing the NanoEdge™ machine learning library

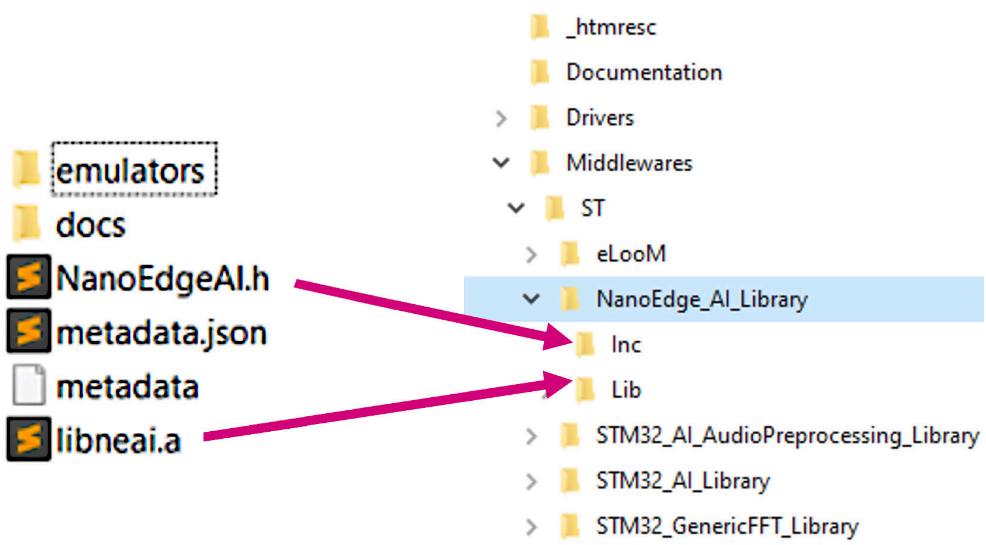
After library generation and download from NanoEdge™ AI Studio, link the libraries to FP-AI-PREDMNT2 and run them on the STWIN.

The FP-AI-PREDMNT2 comes with the library stub in place of the actual libraries generated by NanoEdge™ AI Studio. This allows linking the generated libraries easier.

Step 1. To link the actual libraries, copy the generated libraries and replace the existing stub: `NanoEdgeAI.h` and `libneai.a` included in the `Inc` and `lib` folders, respectively.

The relative path of these folders is `/FP_AI_PREDMNT2/Middlewares/ST/NanoEdge_AI_Library/` as shown in the figure below.

Figure 6. **Path of Inc and Lib folders**



Step 2. Open the project in STM32CubeIDE located in the `/FP-AI-PREDMNT2/Projects/STWINL4R9ZI-STWIN/Applications/PREDMNT2/STM32CubeIDE/` folder and double-click *.project* file (you can choose to use IAR or Keil, instead).

Step 3. To rebuild and install the project on your STWIN board, click on the play button and wait for the "successful download" message.

2.2.3 Testing the NanoEdge™ AI Machine Learning library

Once the STWIN is programmed with the firmware containing a valid library, the condition monitoring libraries are ready to be tested on the sensor board. Now you can issue the learning and detection commands.

To achieve the best results, perform the learning using the same sensor configurations of the contextual data acquisition.

Note: NanoEdge™ AI Studio also offers an offline emulator. It can be very useful to validate the generated library through a validation dataset before embedding it into the FP-AI-PREDMNT2 application.

For further information, refer to the detailed documentation of the NanoEdge™ AI Studio.

2.2.4 Additional parameters and commands in condition monitoring

You can interact with the NanoEdge™ AI library loaded in the application through the official DSH-DASHBOARD.

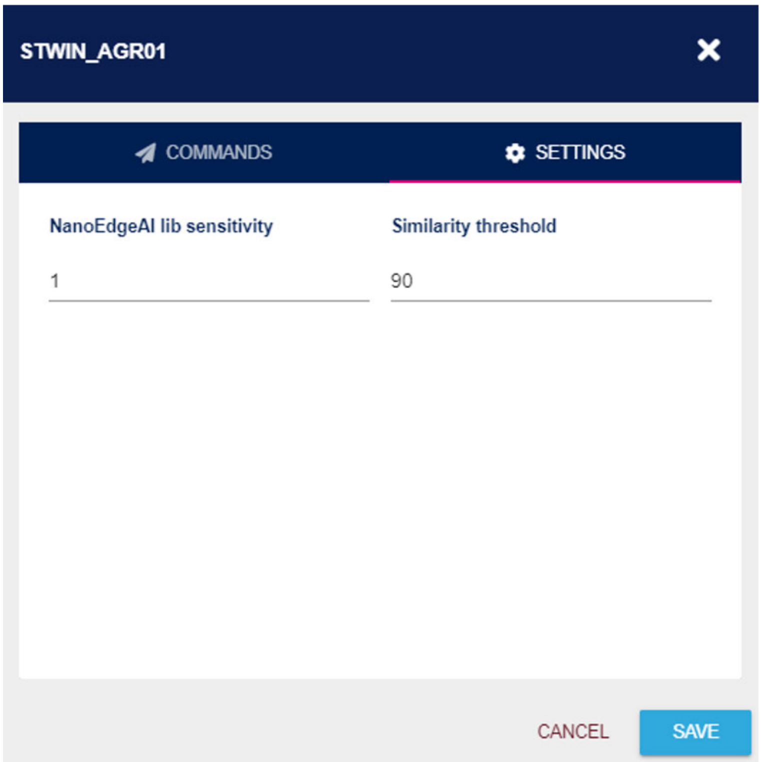
Once the device is connected to the dashboard, the following parameters and commands are available.

Parameters

The sensitivity parameter is used as an emphasis parameter. The default value is set to 1. Increasing this sensitivity means that the signal matching is to be performed more strictly, reducing it will relax the similarity calculation process, i.e. resulting in higher matching values.

The threshold parameter is used to report anomalies. For any signal that has similarities below the threshold value, an anomaly is reported. The default threshold value used in the application is 90. You can change these values in their respective setting boxes.

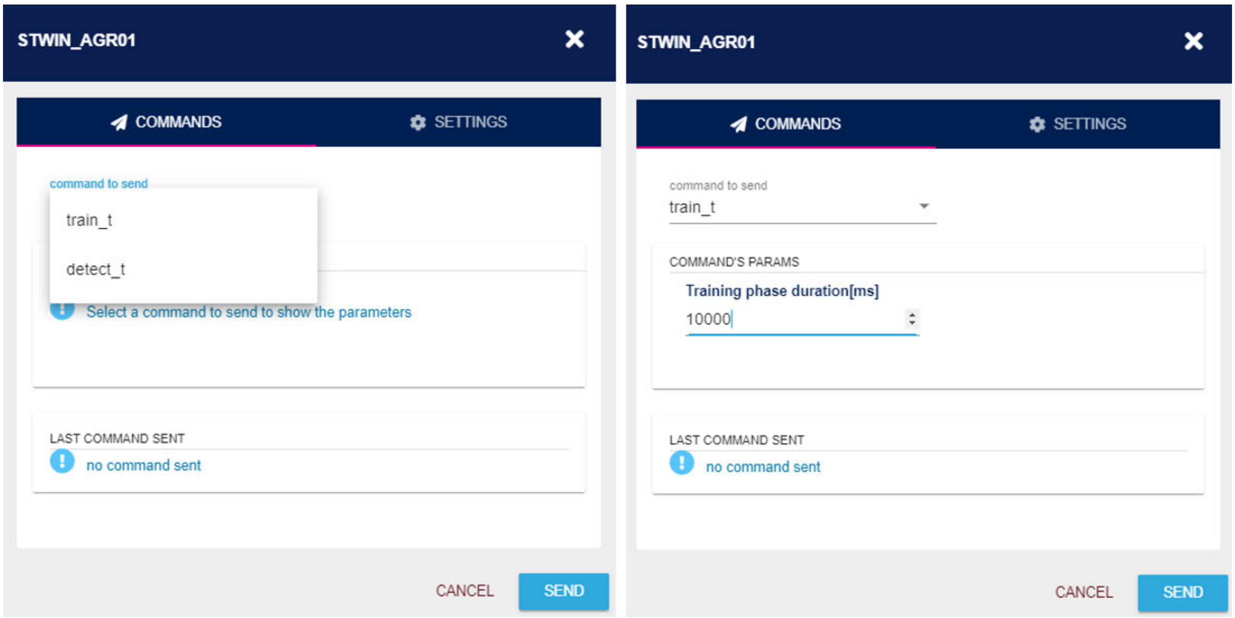
Figure 7. **Parameter setting window**



To better understand the parameters described, refer to the detailed documentation of the NanoEdge™ AI Studio.

Commands

Figure 8. Command setting window



In the “command to send” window you can choose between two commands: "train" or "detect". The result of the train command is the training of the new AI library for the time interval set in the “phase duration” field.

The detect command triggers the anomaly detection phase in the application. In the detection phase if an anomaly is detected, in accordance with the generated library, an anomaly event is sent to the dashboard and an event is shown in the EVENTS panel of the dashboard.

Note: When training or detection phase is active no other output is sent to the dashboard (for example, FFT analysis, environmental data, etc.).

Note: For the best results, expose all the normal conditions to the sensor board during the learning and library generation process, as for example in case of motor monitoring, as well as the required speeds and ramps to monitor.

2.2.5 NanoEdge™ AI library performance

This section describes a specific example of library usage and its performance.

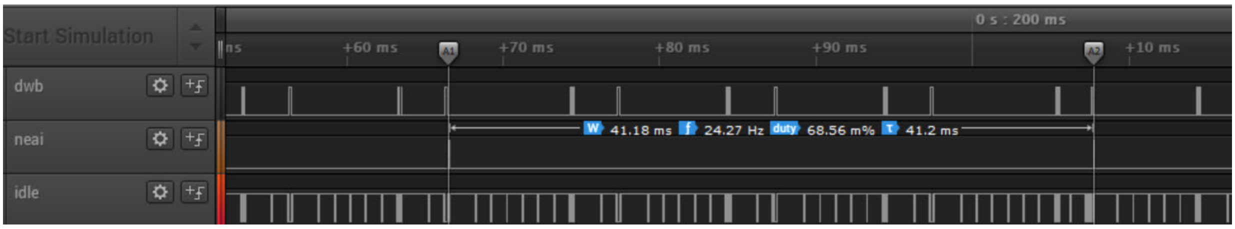
NanoEdge™ AI library processing time must be compatible with respect to the FP-AI-PREDMNT2 CPU usage to avoid real-time issues.

In this example, we consider a library generated with the following parameters:

- Input sensor: IIS3DWBTask
- Axis number: 3
- Signal length: 1024

To avoid any kind of real-time issues and memory overflow, during the NanoEdge AI library processing (inside the NeaiTask), no other signal processing algorithm is performed in the FP-AI-PREDMNT2 demo. Only the sensor tasks remain active during NanoEdge AI processing. In this way, almost the whole MCU processing time is available for the library task.

Figure 9. Time frame to process accelerometer data



The figure above shows the time needed by the generated library to process the requested accelerometer data.

The IIS3DWBTask triggers the NeaiTask each 256 samples. When the NeaiTask has collected 1024 samples from the accelerometer, it starts processing them.

With the parameter described above, each 41.18 ms new data are available to be processed.

The idle time is used here as the placeholder of the time that would be available for the processing task.

Remember: During any execution phase of a NanoEdge AI library, all tasks, except for the sensor ones, are suspended.

In this scenario, we can sum up the idle time inside the library processing interval: the free time available is about 37 ms.

This means that the NeaiTask must finish its processing in 37 ms to prevent any data loss, thus occupying less than 89% (0.89 = 37ms / 41.18ms) of STWIN CPU time.

As a rule of thumb, giving the results obtained from the above measurement, FP-AI-PREDMNT2 application can support a NanoEdge AI custom-generated library requiring at maximum 89% of STWIN CPU (~130 DMIPS out of 150 DMIPS).

2.3 Application configuration and registration

To program the STWIN core system you have to download the relevant version of the STLINK-V3MINI USB driver and follow the procedure below.

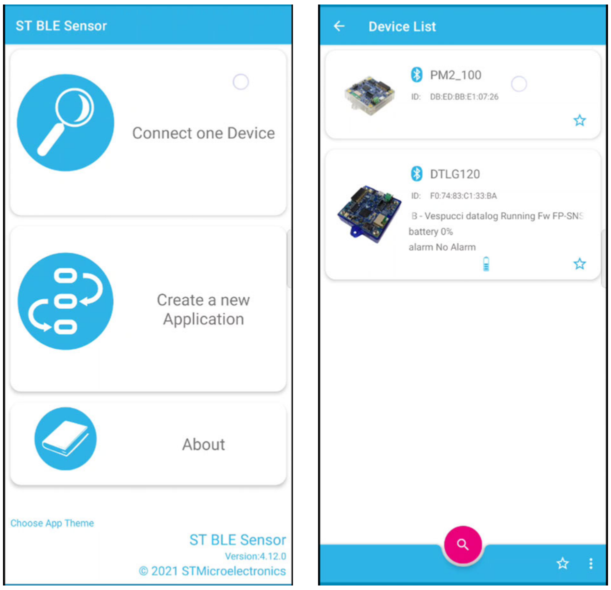
Step 1. Plug the STLINK-V3MINI to the STWIN core board through the programming cable available in the STEVAL-STWINKT1B.

Step 2. Connect the STLINK-V3MINI to the PC through the USB Micro-B male cable.
The board is ready to be programmed with the application in Projects\STM32L4R9ZI-STWIN\Demonstrations\PREDMNT2\Binary\FP-AI-PREDMNT2.bin.

Step 3. Once the firmware has been loaded, press the [**Reset**] black button on the STWIN core system.

Step 4. Open STBLESensor app and connect to PM2_100 device.

Figure 10. Connection to PM2_100



Step 5. Open the menu, select [**Board configuration**] and open [**Certificate Request**].

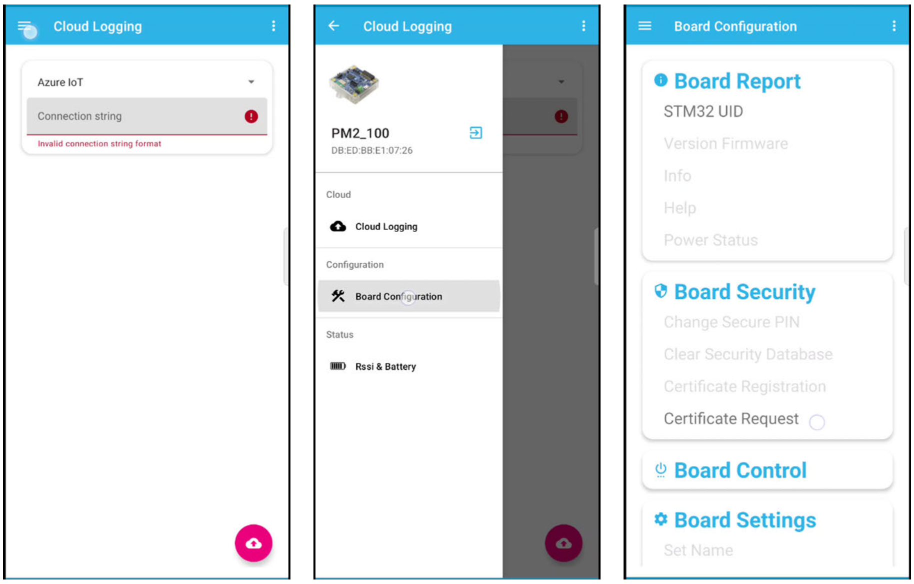


Figure 11. Certificate request

Step 6. Select [**Predictive Maintenance**] and sign in with your MyST credentials.

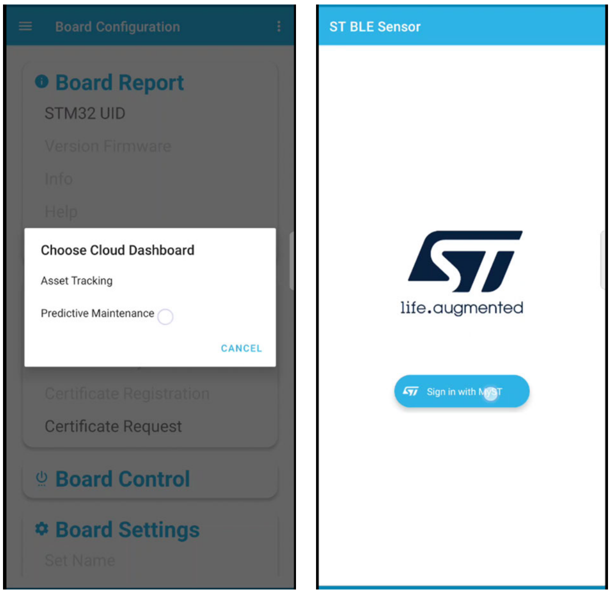
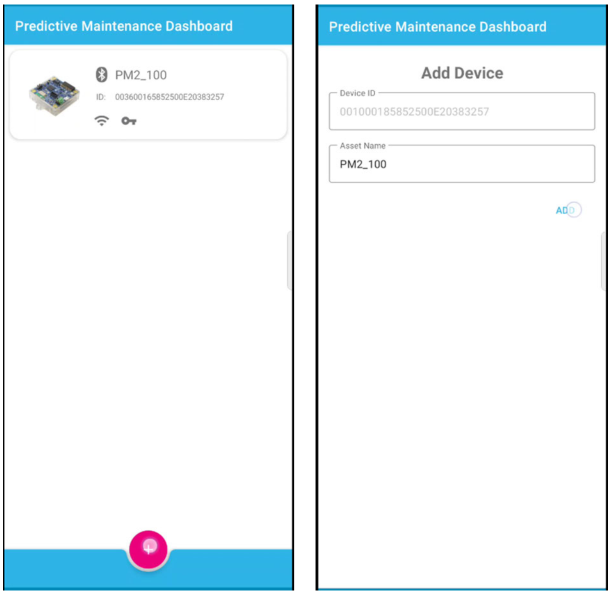


Figure 12. Signing in with your credentials

Step 7. Ask for a new certificate by tapping the + symbol placed at the bottom of the screen or retrieve the ones already associated to your myST account.

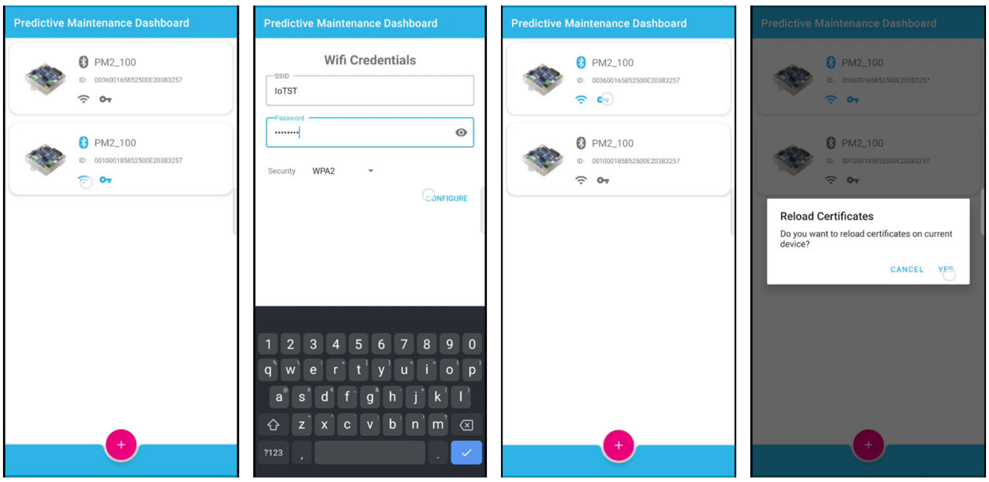
Figure 13. Retrieving a certificate



The certificates valid for your STWIN are highlighted in blue.

Step 8. Tap on the key to download the security certificates into the STWIN and tap on the Wi-Fi symbol to update Wi-Fi credentials.

Figure 14. Downloading security certificate and updating Wi-Fi credentials



If the configuration procedure ends correctly, the orange LED switches on. Now the STWIN is connected to the AWS PREDMNT dashboard and is ready to stream data.

Step 9. Press the user button to start streaming data on the dashboard.

2.4 Application debug

If you want to debug the application, you can recompile it using the preferred IDE in [**Debug configuration**].

The [**Debug configuration**] enables printf and allows streaming debug information from the UART available on the STLINK-V3MINI to a PC terminal.

Step 1. Download and launch a debug session from the IDE.

Step 2. Open a terminal (i.e: in IAR you can use the I/O terminal already available in the IDE, with other IDEs you can launch Tera Term).

Step 3. Set the parameters as shown below:

Figure 15. Tera Term setup (1 of 2)

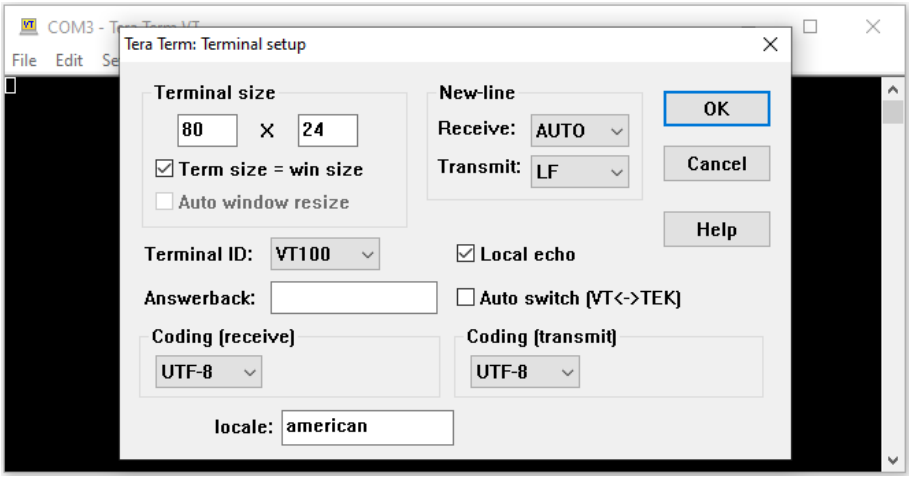
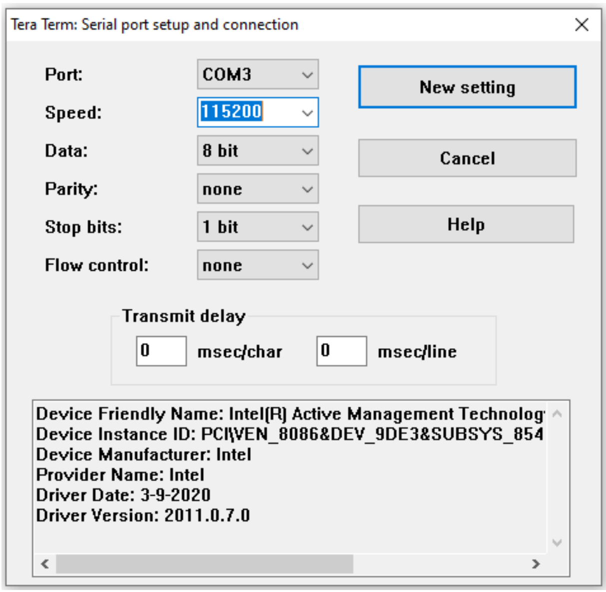
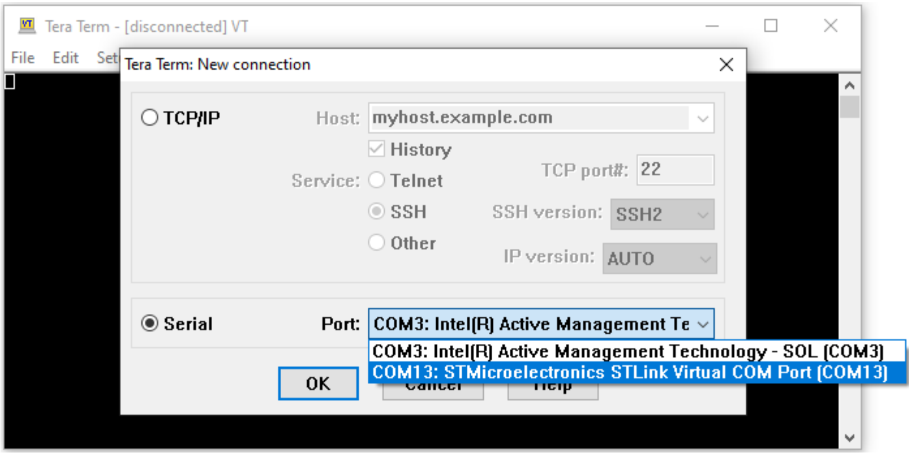


Figure 16. Tera Term setup (2 of 2)



Step 4. Open a new connection and select the ST-LINK COM port.

Figure 17. **Selecting the ST-LINK COM port**



Step 5. Reset the board by pressing the reset button.

2.5 Predictive maintenance dashboard (DSH-PREDMNT)

The DSH-PREDMNT predictive maintenance dashboard is a cloud application based on AWS services.

It provides a highly functional and intuitive interface that is tailored for the collection, visualization, and analysis of condition monitoring data from motion and acoustic vibration sensing elements, as well as temperature and other environmental data.

You can use the dashboard to plot and graph real-time and historical data, monitor critical operating conditions such as running temperature, and set thresholds for automatic warnings when key parameters exceed acceptable limits.

Figure 18. DSH-PREDMNT dashboard overview

- The green button is for:
  - user login and data segregation
  - device and AWS Green Grass Edge
  - registration and configuration:
    - provisioning
    - associations to assets
    - streaming time
    - geo localization
- The light blue button is for:
  - live data visualization
  - add a device to live monitoring
- The pink button is for:
  - asset health monitoring:
    - data collection
    - historical trend analysis
    - failure thresholds settings for alerts and warnings
- The dark blue button is for asset mapping

3 System setup guide

3.1 STEVAL-STWINKT1B wireless industrial node

The STWIN SensorTile wireless industrial node (STEVAL-STWINKT1B) is a development kit and reference design that simplifies prototyping and testing of advanced industrial IoT applications such as condition monitoring and predictive maintenance.

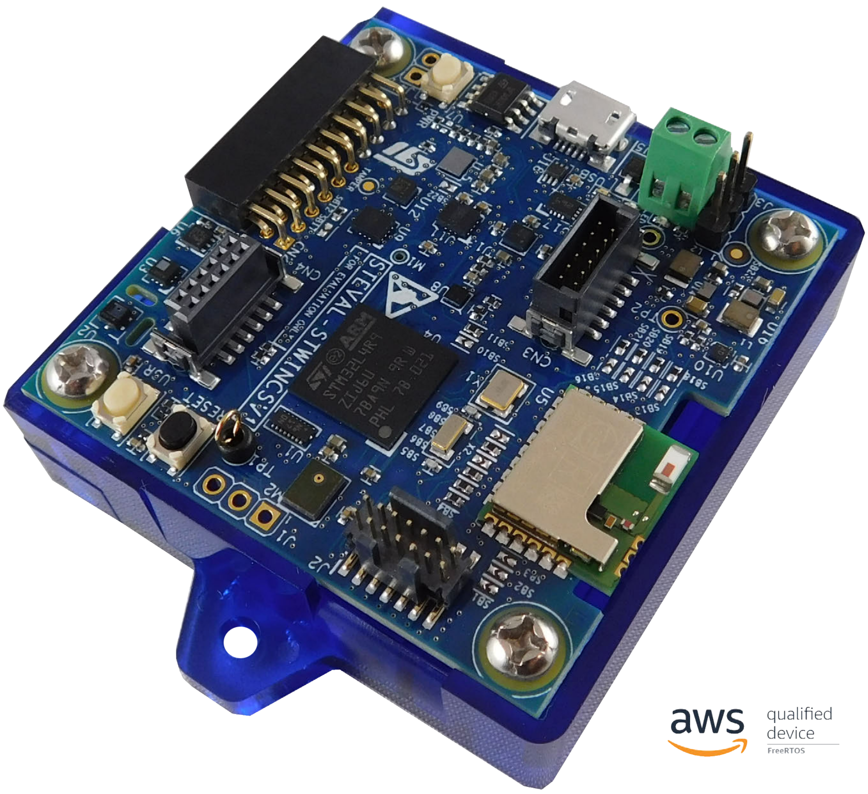
The kit features a core system board with a range of embedded industrial-grade sensors and an ultra-low-power microcontroller for vibration analysis of 9-DoF motion sensing data across a wide range of vibration frequencies, including very high frequency audio and ultrasound spectra, and high precision local temperature and environmental monitoring.

The development kit is complemented with a rich set of software packages and optimized firmware libraries, as well as a cloud dashboard application, all provided to help speed up design cycles for end-to-end solutions.

The kit supports BLE wireless connectivity through an on-board module, and Wi-Fi connectivity through a special plugin expansion board (STEVAL-STWINWFV1). Wired connectivity is also supported via an on-board RS485 transceiver. The core system board also includes an STMod+ connector for compatible, low cost, small form factor daughter boards associated with the STM32 family, such as the LTE Cell pack.

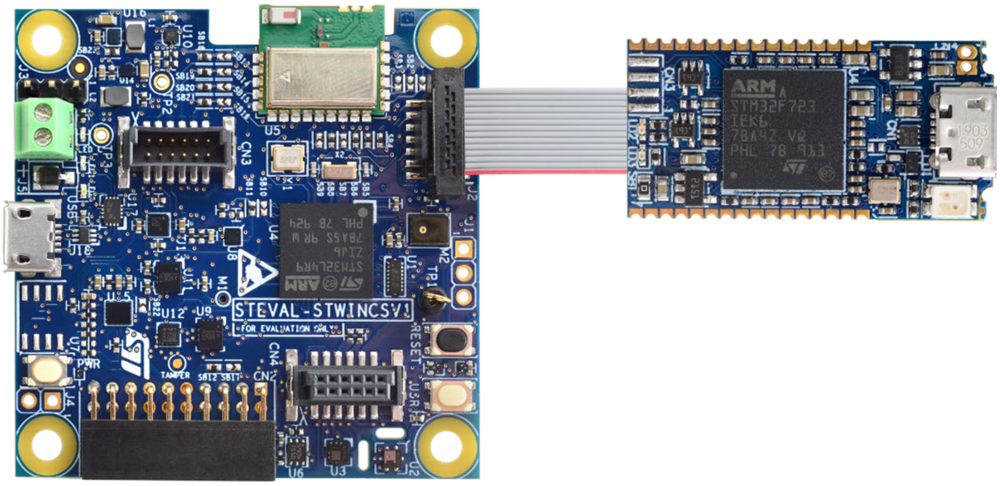
Apart from the core system board, the kit is provided complete with a 480 mAh Li-Po battery, an STLINK-V3MINI debugger and a plastic box.

Figure 19. **STEVAL-STWINKT1B SensorTile Wireless Industrial Node**



The STLINK-V3MINI is a standalone debugging and programming mini probe for STM32 microcontrollers, with JTAG/SWD interfaces for communication with any STM32 microcontroller located on an application board. It provides a virtual COM port interface for host PCs to communicate with target MCUs via UART. The STLINK-V3MINI is supplied with an STDC14 to STDC14 flat cable.

Figure 20. **STEVAL-STWINKT1B connected to STLINK-V3MINI**



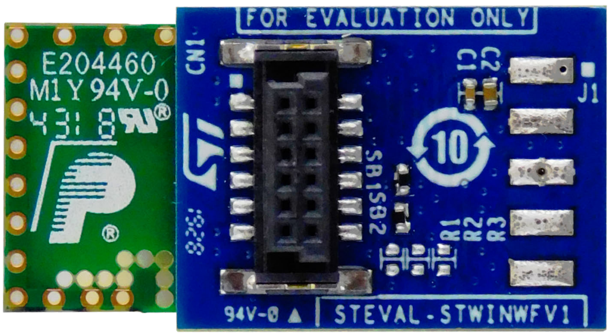
3.2 STEVAL-STWINWFV1

The STEVAL-STWINWFV1 expansion board adds 2.4 GHz Wi-Fi connectivity to the SensorTile Wireless Industrial Network (STWIN) kit (STEVAL-STWINKT1).

It features:

- Wi-Fi adapter for STEVAL-STWINKT1B (and STEVAL-STWINKT1)
- Plugs into STWIN core system board through dedicated 12-pin connector
- Single 3.3 V power supply input
- ISM43362-M3G-L44-E Wi-Fi module:
  – 802.11 b/g/n Compatible based on Broadcom MAC/Baseband/Radio device
  – Fully contained TCP/IP stack
  – Host interface: SPI up to 25 MHz
  – Network features ICMP (Ping), ARP, DHCP,TCP, UDP
  – Low power operation (3.3 V supply) with built-in low power modes
  – Secure Wi-Fi authentication WEP-128, WPA-PSK (TKIP), WPA2-PSK
  – CE certified
  – FCC certified
  – IC certified
  – ARIB certified
- RoHS and China RoHS compliant
- WEEE compliant

Figure 21. **STEVAL-STWINWFV1 expansion board**



3.3 Hardware setup

The following hardware components are needed:

- One STWIN Core System board (order code: STEVAL-STWINKT1B)
- One STLINK-V3MINI debugger
- One STEVAL-STWINWFV1 Wi-Fi expansion board
- Two USB Micro-B cables

3.4 Software setup

The following software components are required for the setup of a suitable development environment to create applications for the STEVAL-STWINKT1B evaluation board:

- FP-AI-PREDMNT2 firmware
- a standard user terminal as Putty or Tera Term (v. 4.97 or higher) only for debugging
- STBLESensor app
- an account for the DSH-PREDMNT predictive maintenance dashboard
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments to select from:
  - IAR Embedded Workbench for ARM® toolchain + ST-LINK
  - RealView Microcontroller Development Kit toolchain + ST-LINK
  - STM32CubeIDE + ST-LINK
- NanoEdge™ AI Studio to generate the condition monitoring libraries.

  NanoEdge™ AI Studio is available at www.st.com/stm32nanoedgeai with several options, such as personalized support-based sprints and trial or full versions.

Revision history

Table 1. Document revision history

Date	Revision	Changes
28-Sep-2021	1	Initial release.
16-Jun-2022	2	Updated introduction, Section 1.1 Overview, Section 1.2 Architecture, and Section 2.2.1.2 Data preparation for library generation with NanoEdge™ AI Studio.

Contents

List of tables

List of figures