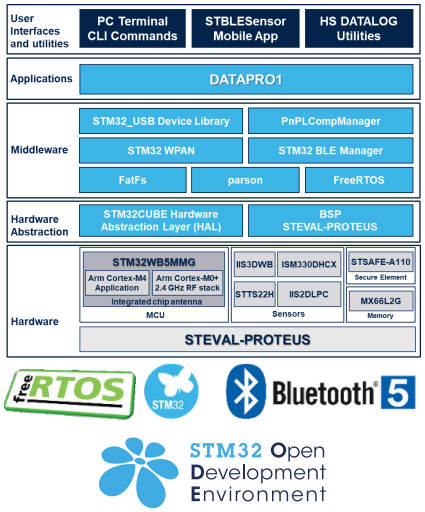
# FP-SNS-DATAPRO1

Data brief

# STM32Cube function pack for STEVAL-PROTEUS1 sensor datalog and file transfer over BLE and USB




## Features

- Firmware package to retrieve and store data from the sensors embedded in the STEVAL-PROTEUS1 evaluation board, using two different options:
  - Wired: on PC folders through USB and PC terminal console
  - Wireless: inside the built-in NOR flash memory controlled by Bluetooth® Low Energy and accessible by USB or Bluetooth® Low Energy
- The wired option is provided by connecting the STEVAL-PROTEUS1 to a PC via USB, configured with high speed class (USB WCID), plus a customized data capture software suite.
- The wireless option is built around the data storage on embedded NOR flash memory controlled and accessible by STBLESensor app, and accessible by USB mass storage class (USB MSC).
- Compatible with STBLESensor (Android and iOS), for sensor settings, system monitoring, and FUOTA.
- Embedded software, middleware, and drivers:
  - FreeRTOS™ third-party RTOS kernel for embedded devices
  - FatFS: third-party for FAT file system module for small embedded systems
  - PnPLCompManager to handle PnP-like commands and properties generated through a digital twin definition language (DTDL)
  - STEVAL-PROTEUS1 BSP drivers

| Product summary | |
|---|---|
| STM32Cube Function Pack for STEVAL-PROTEUS1 Sensor Datalog and File Transfer over BLE and USB | FP-SNS-DATAPRO1 |
| Industrial sensor evaluation kit for condition monitoring based on the 2.4 GHz STM32WB5MMG module | STEVAL-PROTEUS1 |
| BLE sensor application for Android and iOS | STBLESensor |
| Applications | Conditon Monitoring & Predictive Maintenance<br>IoT for Smart Industry |

## Description

The FP-SNS-DATAPRO1 is a STM32Cube function pack for the STEVAL-PROTEUS1 that provides a data logging of the embedded sensors, supporting wired or wireless connectivity.

Sensor data can be stored into the embedded NOR flash memory formatted as FatFs or streamed to a PC via USB (WCID device).

Offline sensor data analysis can be performed on stored data by dedicated HSDPython SDK utilities as already included in other firmware packages.

The application is compatible with the STBLESensor mobile app to configure sensor parameters and file data transfer.

A JSON configuration file can be stored inside the NOR flash memory by means of the STBLESensor mobile app, or directly copied using the STEVAL-PROTEUS1 as USB mass storage. The application uses this configuration file during any starting phase.

The data logging in the wired option is based on USB connectivity supported by Python and C++ real-time control applications.

Sensor data acquisition through a few easy-to-use scripts in Python (Rel. 3.7 or above), provided within the software package.

The data capture software suite is executed by a set of CLI commands on a PC terminal that allow to configure sensors as well as start/stop high speed data logging, and its automatic storage in a PC's folder.

In alternative, the wireless option is based on the STBLESensor app (available for Android from v 5.2 and above), where the user can manage the board and sensor configurations, start/stop data acquisition. The sensor data is stored into embedded NOR flash memory formatted as FatFs.

During this option is also possible to use an automode functionality, that enable start and stop a sequence of datalogs, configurable through a configuration JSON file or directly using the STBLESensor mobile app.

File data transfer can be implemented in two different ways. STBLESensor app, allow to select and send via Bluetooth each acquisition data file from STEVAL-PROTEUS1 to the mobile device.

Connecting the STEVAL-PROTEUS1, as USB mass storage, to a PC, you can use a file manager to browse inside the built-In NOR flash memory in read and write mode.

# 1 Detailed description

## 1.1 What can you do with STM32Cube function packs?

STM32Cube function packs leverage the modularity and interoperability of STM32 Nucleo and X-NUCLEO boards together with STM32Cube and X-CUBE software to create function examples for some of the most common use cases of different application technologies.

These software function packs are designed to exploit the underlying STM32 ODE hardware and software components as much as possible to best satisfy the requirements of final user applications.

Moreover, function packs may include additional libraries and frameworks that are not present in the original X-CUBE packages, thus enabling new functionalities allowing real and usable system for developers.

## 1.2 What is STM32Cube?

STM32Cube is a combination of a full set of PC software tools and embedded software blocks running on STM32 microcontrollers and microprocessors:

- STM32CubeMX configuration tool for any STM32 device; it generates initialization C code for Cortex-M cores and the Linux device tree source for Cortex-A cores
- STM32CubeIDE integrated development environment based on open-source solutions like Eclipse or the GNU C/C++ toolchain, including compilation reporting features and advanced debug features
- STM32CubeProgrammer programming tool that provides an easy-to-use and efficient environment for reading, writing and verifying devices and external memories via a wide variety of available communication media (JTAG, SWD, UART, USB DFU, I2C, SPI, CAN, etc.)
- STM32CubeMonitor family of tools (STM32CubeMonRF, STM32CubeMonUCPD, STM32CubeMonPwr) to help developers customize their applications in real-time
- STM32Cube MCU and MPU packages specific to each STM32 series with drivers (HAL, low-layer, etc.), middleware, and lots of example code used in a wide variety of real-world use cases
- STM32Cube expansion packages for application-oriented solutions.

## 1.3 How does this function pack complement STM32Cube?

This software is based on the STM32CubeHAL. It extends STM32Cube by providing a board support package (BSP) for the STEVAL-PROTEUS1 evaluation kit.

The drivers abstract low-level details of the hardware and allow the middleware components and applications to access data in a hardware-independent manner.

The package includes some middleware libraries to store data onto a built-in NOR flash memory (through a third-party FatFS module) and stream data to a PC via USB (thanks to the WCID USB class).

The application also takes advantage of the FreeRTOS™ module, thus enabling a real-time operating system into an STM32 microcontroller.

To cover the STM32WB Bluetooth® Low Energy embedded capabilities, the package includes also the powerful of STM32_WPAN middleware.

## Revision history

Table 1. **Document revision history**

| Date | Version | Changes |
|---|---|---|
| 11-Dec-2023 | 1 | Initial release. |
| 27-Feb-2024 | 2 | Updated Section Description. |

**IMPORTANT NOTICE – 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 acknowledgment.

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. For additional information about ST trademarks, refer to www.st.com/trademarks. 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.

© 2024 STMicroelectronics – All rights reserved