STSW-SDATALOG
Quick Start Guide

Version 1.0 – Dec ‘25

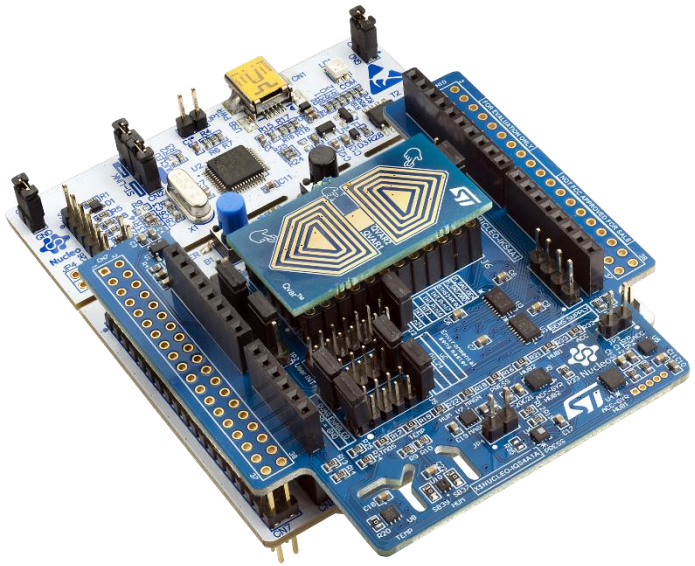
Overview

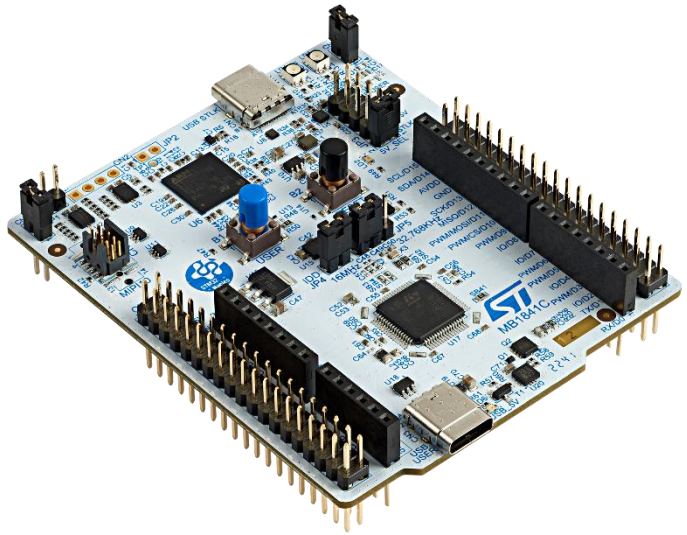
- The STSW-SDATALOG provides simple serial data logging firmware implementations.
- It allows you to easily stream sensor data to a host PC via a USB serial VCOM port in real time.
- The software is a collection of bare-metal firmware examples designed to generate well-formed datasets compliant with the FP-SNS-DATALOG2 data format.
- It is supported by the STDATALOG-PYSDK, offering portability across custom boards and compatibility with several example boards.

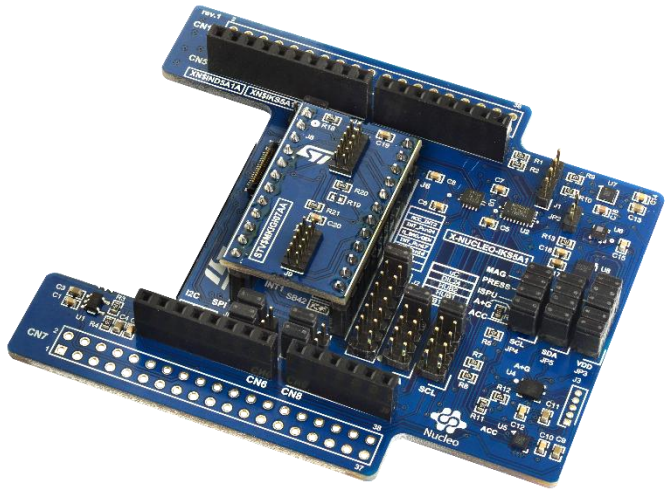
Hardware setup

Supported Hardware

- The STSW-SDATALOG contains examples for NUCLEO-F401RE, NUCLEO-L476RG, and NUCLEO-U545RE-Q with X-NUCLEO-IKS4A1 and X-NUCLEO-IKS5A1.
- No hardware modifications are required: simply plug the X-NUCLEO on top of the NUCLEO board and flash the firmware
- Only a USB cable is needed: Type-C for NUCLEO-U545RE-Q, and micro-USB for the others.



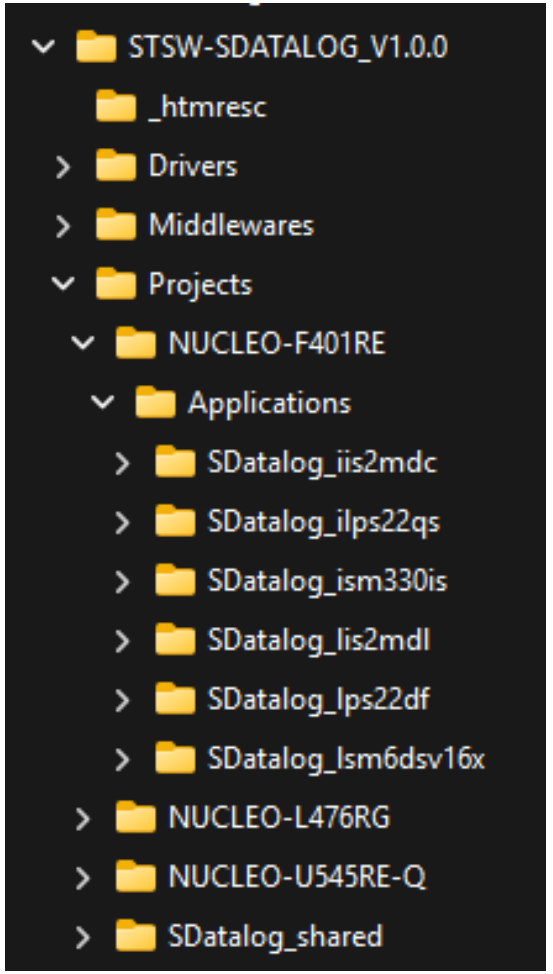




Software setup

Serial Datalog

- STSW-SDATALOG provides a simple and bare-metal data logging firmware solution, focusing on single sensor dataset generation.
- Data streams include precise timestamping and control communication protocol is compliant with format adopted by FP-SNS-DATALOG2. So, the well-formed produced datasets are natively compatible with STDATALOG-PYSDK.
- Developed exclusively in STM32CubeIDE, each example supports a single sensor. Specialized examples for different sensor typologies are available: accelerometer, combo (accelerometer + gyroscope), magnetometer, and barometer.
- To simplify portability and code reusability, most of the code is reusable and shared among different examples. Furthermore, all firmware configurations are managed using STM32CubeMX: *.ioc* files are provided for each example.



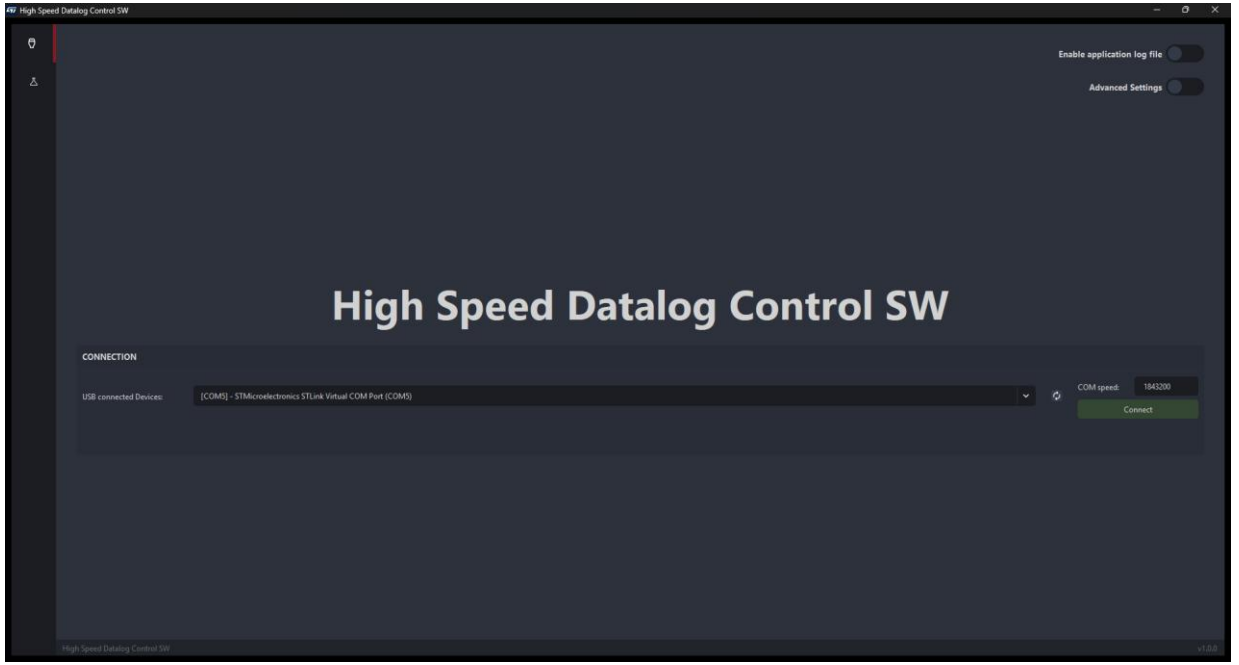
SDATALOG with STDATALOG-PYSDK

- SDATALOG examples are natively compatible with STDATALOG-PYSDK
- The STDATALOG-PYSDK is a comprehensive Python framework designed to facilitate the capture, processing, and visualization of data from a wide range of sources, including sensors, algorithms, simulated signals, and telemetry from actuators.
- The python software development kit (SDK) for data logging has been developed using Python 3.13.
  - It is compatible with Python 3.10, 3.11, and 3.12 also.
  - Python must be already properly installed on the user's machine before installing and running STDATALOG-PYSDK.
  - *STDATALOG-PYSDK* requires various Python modules. The package is distributed with *installers* that resolve all the required dependencies
- The next slides show how to use the *stdatalog_GUI.py* script available in the SDK. Please see the full documentation available on the STDATALOG-PYSDK landing page to install the SDK properly on your machine

Execute *stdatalog_GUI.py*

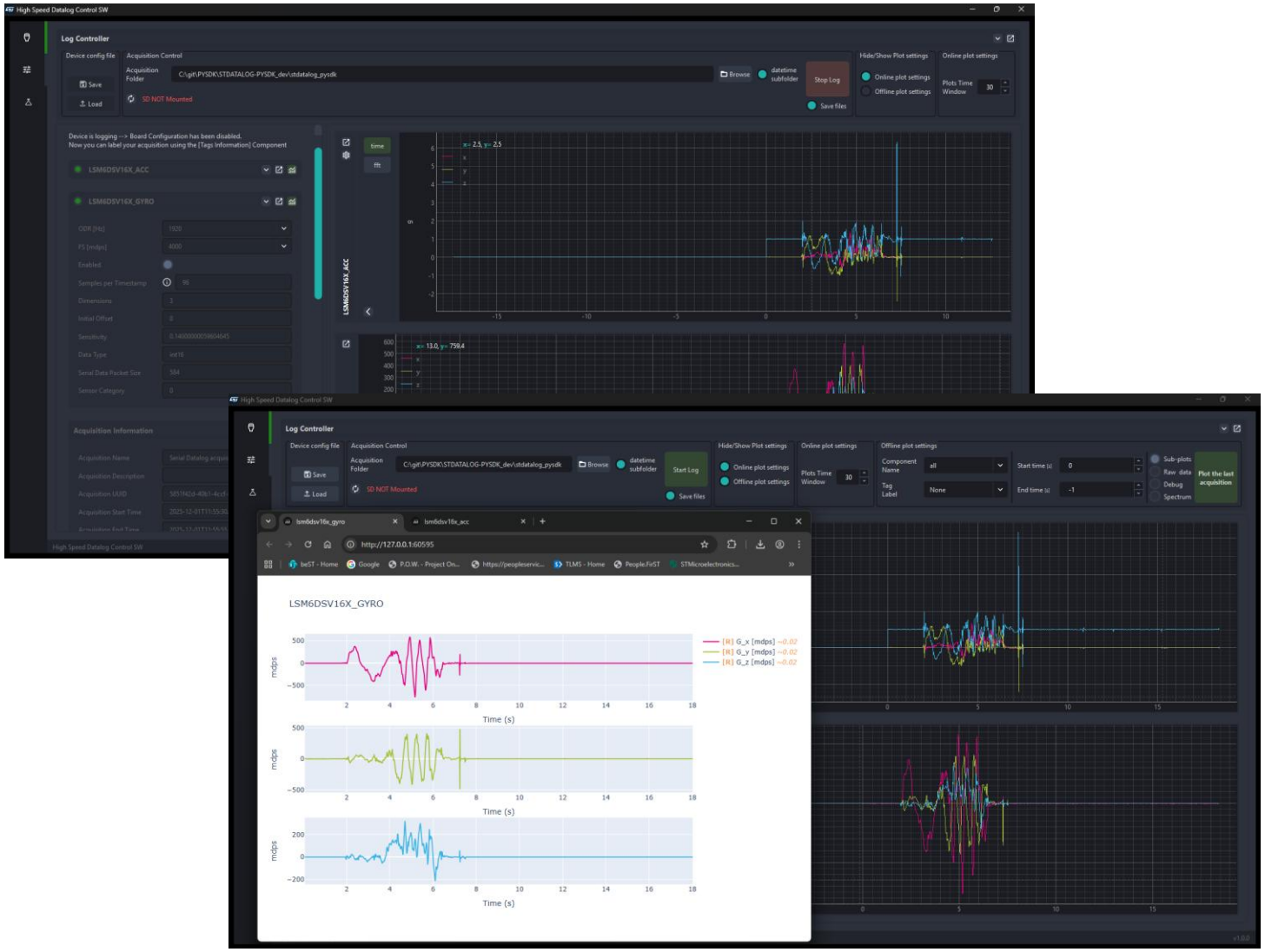
- *stdatalog_GUI.py* works within the STDATALOG-PYSDK, developed in Python 3.13 on Windows, Linux and macOS environments.
  - *STDATALOG-PYSDK* requires different Python modules. The package is distributed with *installers* that solve all the required dependencies
  - Please see the full documentation available on the STDATALOG-PYSDK landing page to install the SDK properly on your machine

- Once the board is connected via USB and the Python environment is properly updated, you can launch the real-time plot by executing *stdatalog_GUI.py* available in *stdatalog_examples\gui_applications\stdatalog\GUI*.
  - Depending on your local setup, to execute the script, you can open a command shell there and run *python stdatalog_GUI.py*.
- Click on the Connect button to allow the connection between the board and the PC.



stdatalog_GUI.py

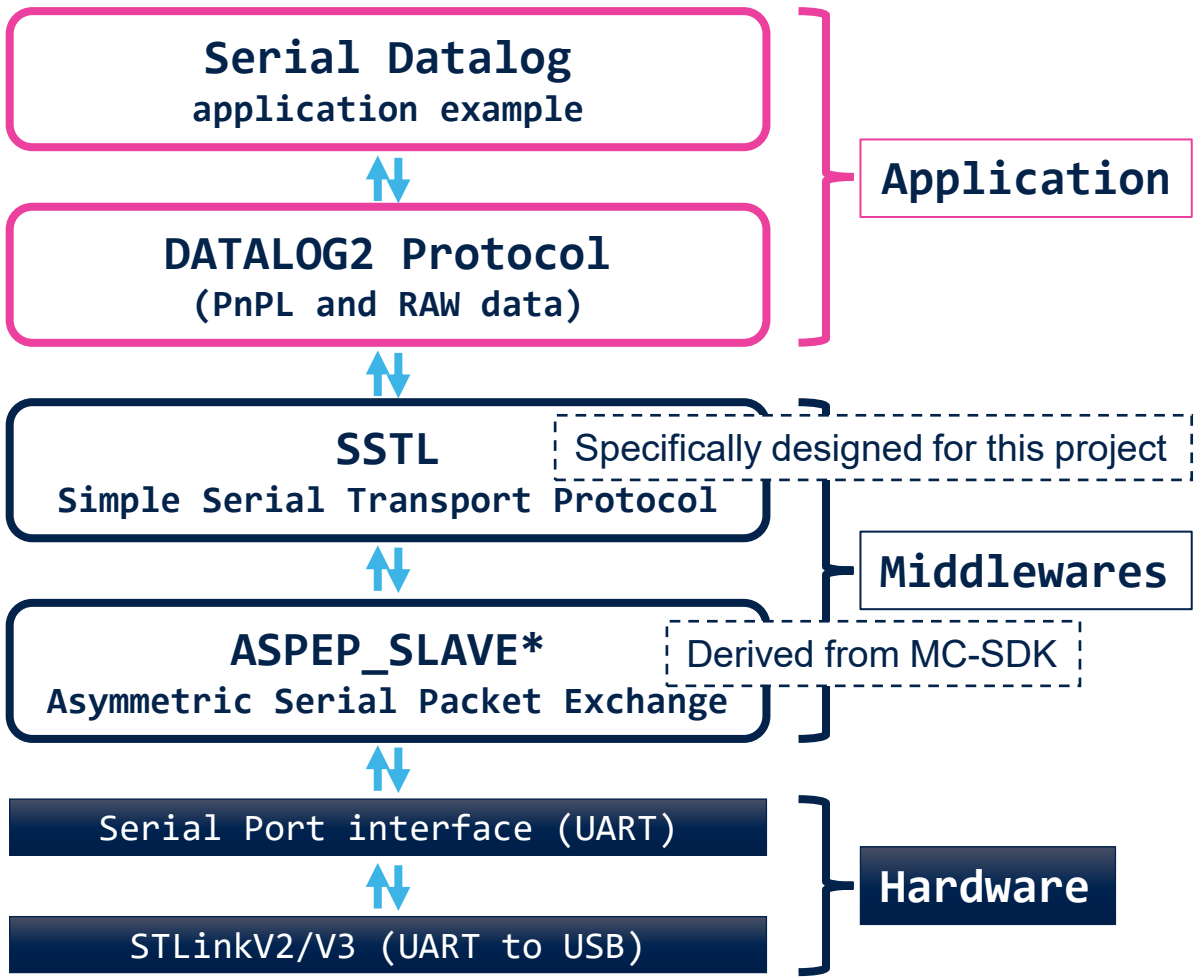
- Once the connection is established, you can:
  - Enable or disable the needed sensors.
  - Set up data rate, full scale, and timestamps.
  - Retrieve sensor status.
  - Start or stop logging data on the PC.
- Once you click the *“Start Log”* button, data are live plotted, and the application creates a YYYYMMDD_HH_MM_SS (e.g., 20200128_16_33_00) folder containing the raw data and the JSON configuration file.
- Once the test is concluded, you can also check and plot the entire dataset by clicking on the *"Plot the last acquisition"* button.



Firmware design

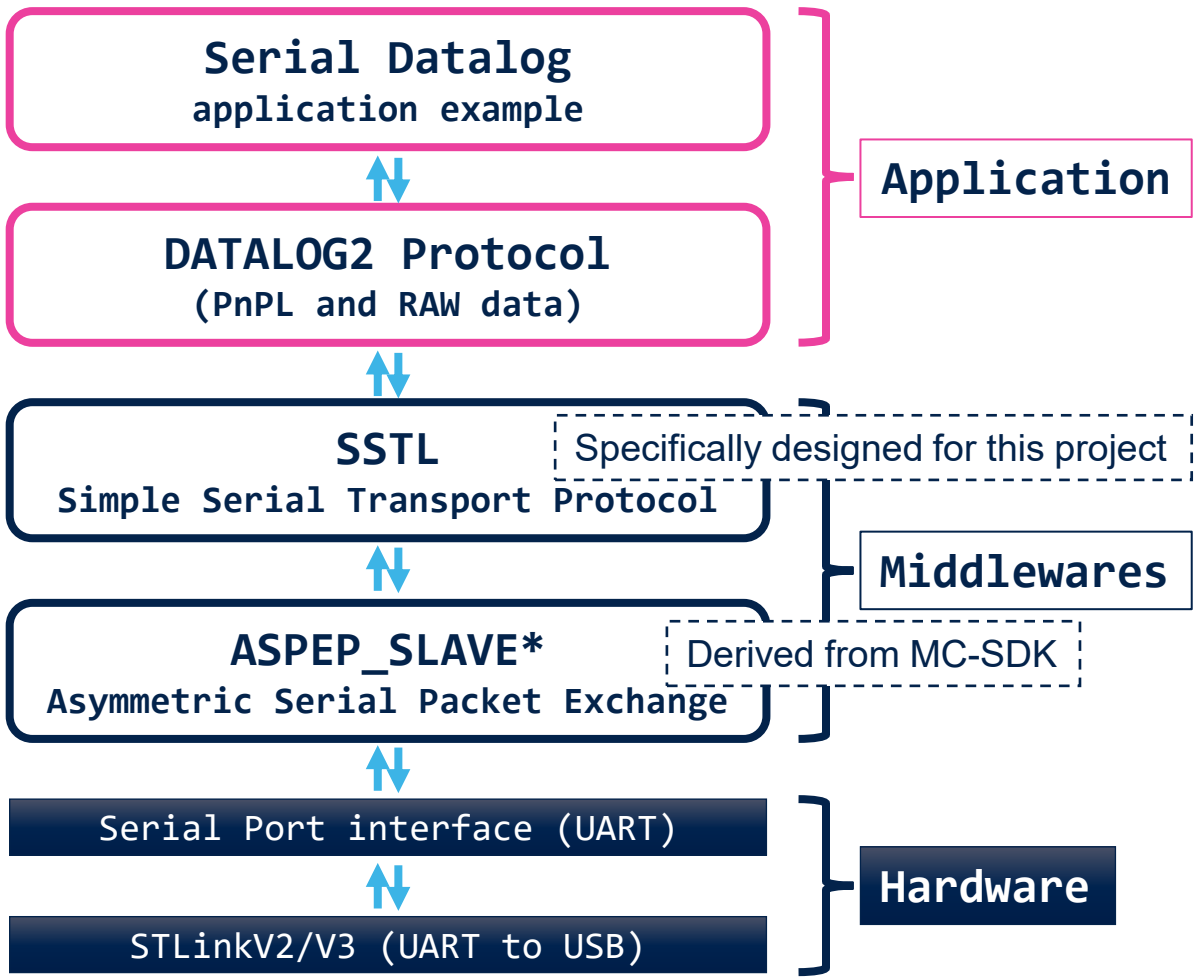
FW Application stack

- Code portability and reusability are one of the main goals of STSW-SDATALOG design
- On top of the application code specific for each sensor/example (PnPL files, main, sensor_app, …) protocols and methods are developed in a common shared folder called SDatalod_shared
- sdatalog.c file is the main entry point for the Serial Datalog application. It is platform independent and can be used with any sensor or STM32.
  - Initializes the sensor API, HW timer for timestamp, and serial protocol.
  - Initializes the PnPL components and sets up the datalog stream parameters.
  - Manages incoming messages, runs a state machine, and reads sensor data if streaming is active and data is ready.
  - Handles buffer management, updates the byte counter and manages sensor data streaming via UART using SSTL protocol



FW Application stack

- SSTL is a simple point-to-point protocol specifically designed to transfer packets on a serial link, like a UART peripheral. It's a master/slave protocol that supports both synchronous and asynchronous packets, and it's tailored specifically to multi-channel data streaming applications. readme.html file available in Middlewares/ST/SimpleSerialTL describes protocol details.
- SSTL implementation is built on top of the ASPEP protocol layer, a serial transport protocol developed in Motor Control SDK. SDATALOG examples implement ASPEP Slave, while Master role is implemented on host side by STDATALOG-PYSDK. For a complete reference of the ASPEP protocol, see MCSDK documentation. Official documentation on ASPEP protocol can be found in X-CUBE-MCSDK installation folder also.



Our technology starts with You

Find out more at www.st.com