How to use the software package for the SensorTile Wireless Industrial Node based on STM32Cube

Introduction

The STSW-STWINKT01 firmware package for the SensorTile Wireless Industrial Node (STWIN) development kit provides sample projects that you can use to develop custom predictive maintenance, smart industry, IoT and remote monitoring applications. The package is based on STM32Cube software technology, and includes all the low level drivers to manage the on-board devices and system-level interfaces.

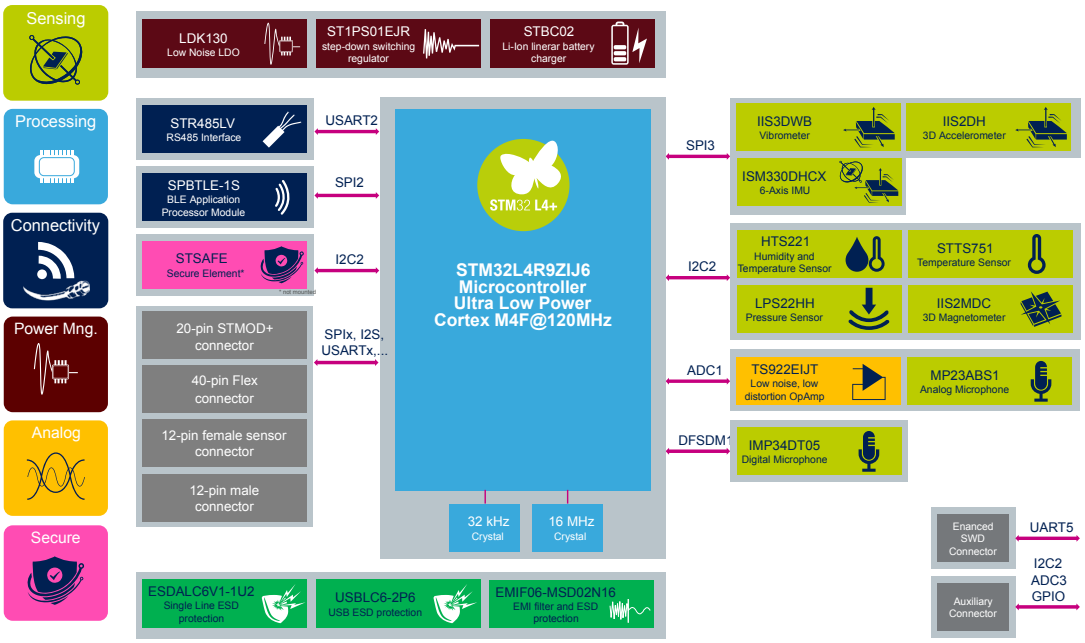
The package features one project that demonstrates data logging functionality. It involves streaming data via the USB Virtual COM Port class, which you can subsequently display directly on a PC terminal.

Other two projects demonstrate wireless connectivity using Bluetooth and Wi-Fi. The Bluetooth project allows you to stream environmental sensor data via the Bluetooth Low Energy (BLE) protocol, and is compatible with our freely available ST BLE Sensor app on Android and iOS stores, so you can read and manipulate the data from your mobile device. The Wi-Fi project requires the STEVAL-STWINWFV1 Wi-Fi expansion board (not included in the kit) to implement basic network functionality including pinging a remote station, connecting to a TLS secure server, sending data to an echo server and verifying returned data, and running a server that a remote client can connect to.

For HSDatalog example, refer to FP-SNS-DATALOG1 software package.

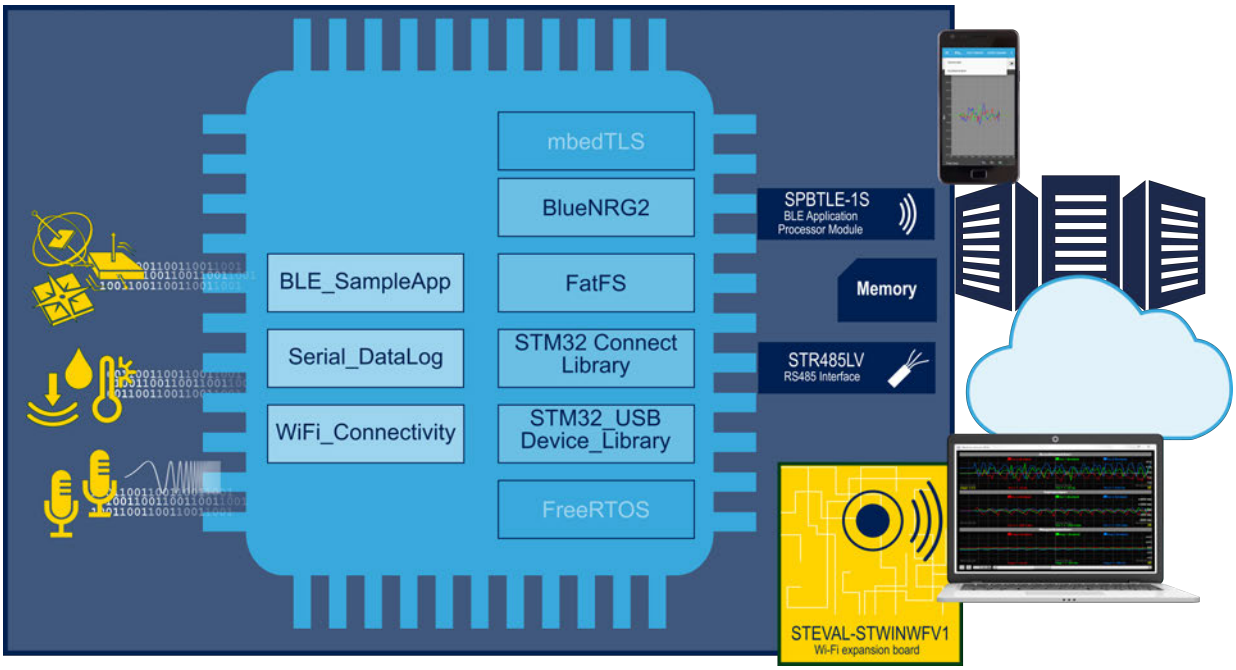
For audio streaming examples, refer to X-CUBE-MEMSMIC1 software package.

Figure 1. SensorTile Wireless Industrial Node application diagram



1 Overview

Figure 2. **STSW-STWINKT01 software block diagram**



The STSW-STWINKT01 firmware expands the functionality of the STM32Cube environment with many features and functions needed to build wireless predictive maintenance and condition monitoring applications and provides some working code examples to help you become familiar with the underlying middleware.

The key features of the package are:

- Set of firmware examples that show how to implement basic functions on the STWIN (STEVAL-STWINKT1) kit:
  - Sensor data streaming example via USB terminal (VCP)
  - Wi-Fi network functionality using the connectivity framework (with Wi-Fi expansion board)
  - Sensor data streaming via BLE
  - Source code freely available from the ST website with developer-friendly license terms
- Embedded software, middleware and drivers:
  - FatFS third party FAT file system module for small embedded systems
  - FreeRTOS third party RTOS kernel for embedded devices
  - STWIN Low-Level BSP drivers
- Based on STM32Cube software development environment for STM32 microcontrollers

2 How to program the board

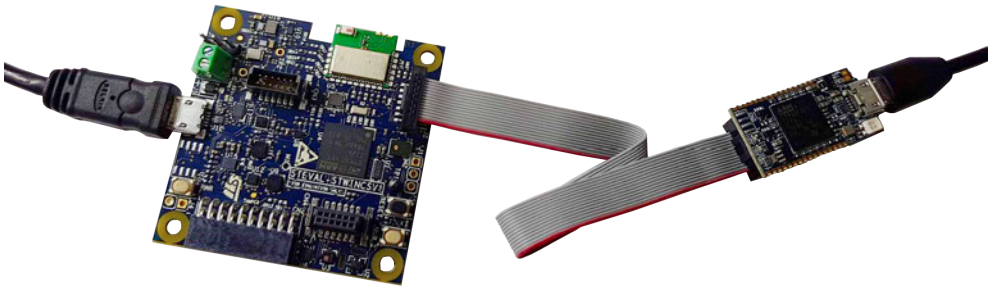
2.1 How to program STWIN with STLINK-V3MINI

Follow the procedure below to program the STWIN core system board.

Step 1. Connect the STWIN core system board to the STLINK-V3MINI programmer using the 14-pin flat cable. The programmer and the cable are included in the hardware kit.

Step 2. Connect both the boards to a PC using micro USB cables.

Figure 3. STLINK-V3MINI connected to STWIN core system board



Step 3. Download the firmware onto the core system board; you can either:

- download one of the sample application binaries provided using STM32CubeProgrammer or ST-LINK Utility
- recompile one of the projects with your preferred IDE (EWARM, Keil, STM32CubeIDE)

2.2 How to program STWIN without STLINK-V3MINI using STM32CubeProgrammer "USB mode"

The can also be reprogrammed via USB using the STM32CubeProgrammer "USB mode".
To enter "Firmware upgrade" mode you must follow the procedure below:

Step 1. Unplug the STWIN core system board.

Step 2. Press the USR button.

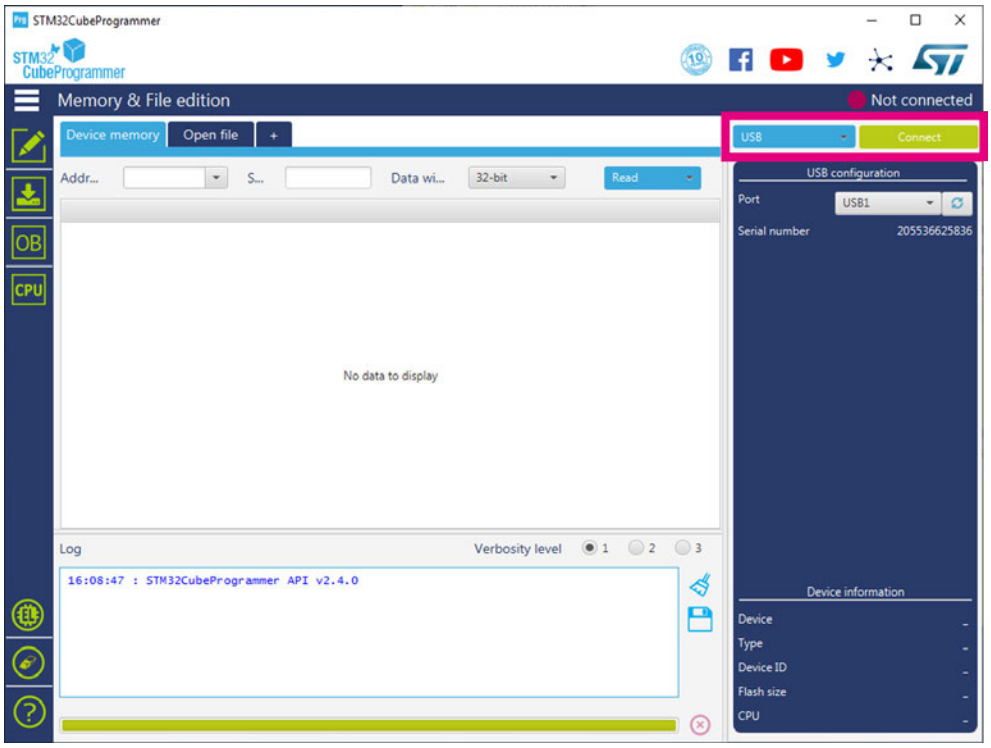
Step 3. While keeping the button pressed, connect the USB cable to the PC.
Now the board is in DFU mode.

Step 4. You can upgrade the firmware by following the steps below:

Step 4a. Open STM32CubeProgrammer.

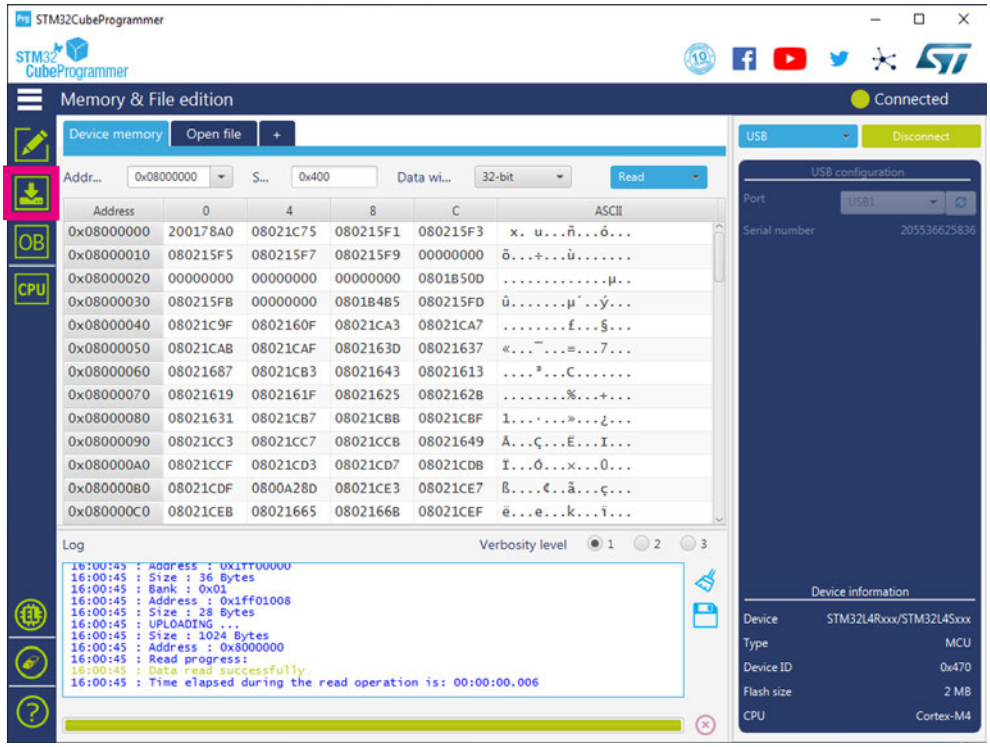
Step 4b. Select [**USB**] on the top-right corner.

Figure 4. STM32CubeProgrammer - USB mode selection



Step 4c. Click on [**Connect**].

Figure 5. STM32CubeProgrammer - connection

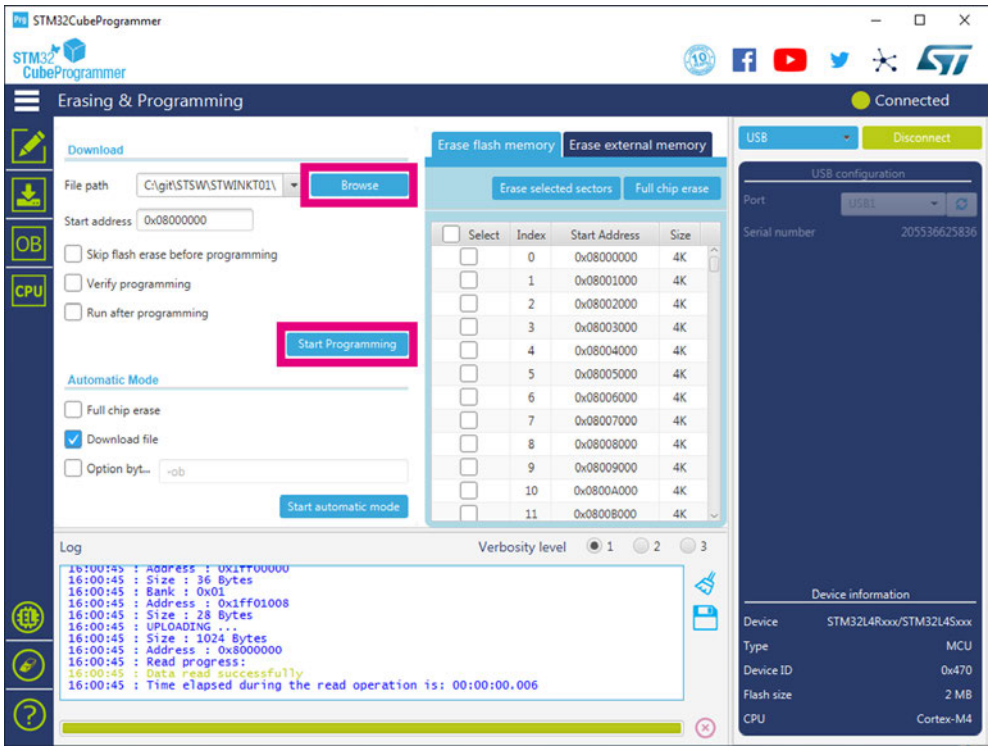


Step 4d. Go to the [**Erasing & Programming**] tab.

Step 4e. Search for the new .bin or .hex binary file to be flashed into the board.

Step 4f. Click on [**Start Programming**].

Figure 6. STM32CubeProgrammer - programming



3 Sample applications

The STSW-STWINKT01 firmware package includes applications to stream sensor data via USB Virtual COM Port class, use Wi-Fi connection and Bluetooth Low Energy configuration.

RELATED LINKS

For audio sample applications (OnboardMics, UltrasoundFFT and MicArrayCoupon), refer to X-CUBE-MEMSMIC1

For HSDatalog sample application refer to FP-SNS-DATALOG1

3.1 Serial_DataLog application

The Serial DataLog application shows how to stream sensor data via USB Virtual COM Port class, so the data can be viewed using any serial terminal software like TeraTerm or PuTTy.

After reset, the firmware performs the following initial tasks:

1. configures HAL and clocks
2. configures LED1 and LED2
3. initializes the USB peripheral
4. creates the threads and activate FreeRTOS scheduler

The `GetData_Thread` and `WriteData_Thread` threads are scheduled by FreeRTOS with different priorities and communicate with each other through a message queue:

- `GetData_Thread`: a high priority task that configures the sensors, reads data at a given frequency and pushes new data in the queue. An OS timer triggers the execution of the thread at a given frequency.
- `WriteData_Thread`: a low priority task that writes sensor data as soon as they are available in the queue.

After you have downloaded the firmware, you can follow the procedure below to run the Serial DataLog application:

Step 1. Connect the board to a PC via micro-USB cable.
The PC will recognize the board as a Virtual COM Port.

Step 2. Open the COM port with a serial terminal like Putty or TeraTerm.
Use the following parameters: 8N1, 115200 bauds, no HW flow control, line endings LF or CR-LF (Transmit) and LF (receive).
When connected to the PC, the board configures the sensors and starts streaming data to the PC.

RELATED LINKS

Appendix A Serial Terminal configuration using TeraTerm on page 10

If required, you can download the Windows driver from the ST website: VCP driver

3.2 WiFi_Connectivity application

This sample application requires STEVAL-STWINWFV1 Wi-Fi expansion plugged on the CN3 connector of the core system board.

Figure 7. STEVAL-STWINWFV1 expansion on core system board CN3 connector

The application implements basic network functionality using the connectivity framework.

`WiFi_Connectivity` demonstrates how to set up the `STM32_Connect_Library` Middleware to allow the STWIN core system board to:

- Ping a remote station
- Connect to a TLS secure server without server identification check
- Connect to a TLS secure server with server identification check
- Send data to an echo server and check return data
- Run a Server, waiting for connection from a remote client

Following reset, the firmware performs the following initial tasks:

1. configures HAL and clocks
2. configures LED1 and LED2
3. configures USB peripheral
4. configures the network interface and initialize the Wi-Fi module
5. starts the connectivity examples

After you have downloaded the firmware, you can follow the procedure below to run the WiFi Connectivity application:

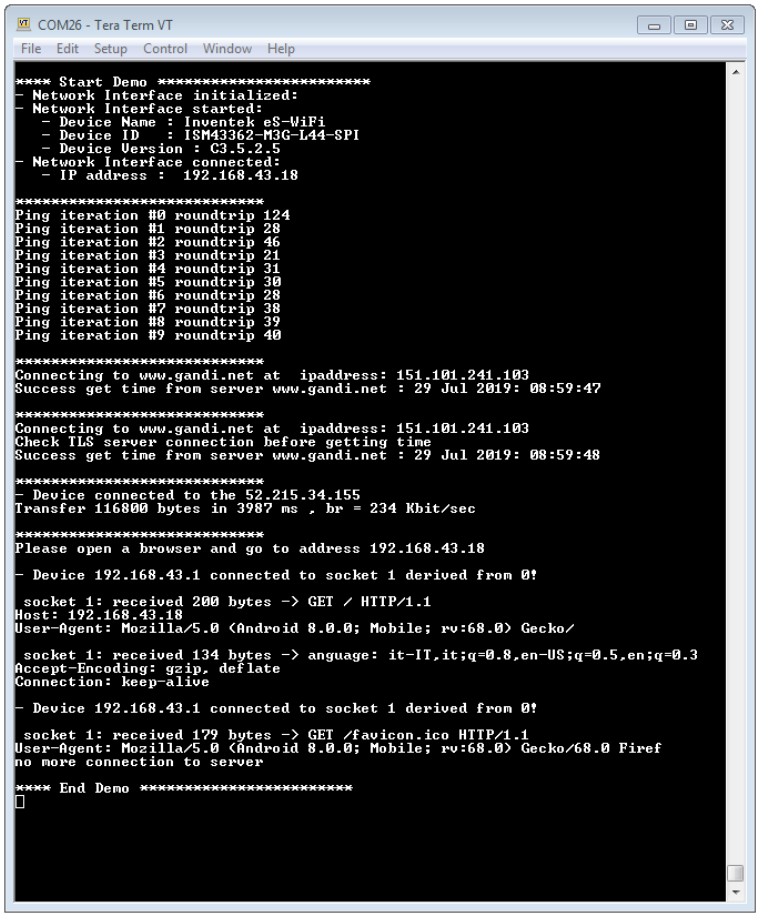
Step 1. Connect the core system board with Wi-Fi module expansion board to a PC with a micro-USB cable.
The PC will recognize it as a Virtual COM Port.

Step 2. Open the COM port with a serial terminal like Putty or TeraTerm.
Use the following parameters: 8N1, 115200 bauds, no HW flow control, line endings LF or CR-LF (Transmit) and LF (receive).

Step 3. Run the program.

Your terminal console will return output status information.

Figure 8. Serial terminal output for Wi-Fi_Connectivity app status



Step 4. Change default IPs and access credentials (optional).

By default, the STWIN tries to connect to the following Wi-Fi network:

- SSID: stwintest
- Password: stwintest
- Security: WPA2 PSK
- Band: 2.4GHz

The default SSID and Password may be changed by modifying the following defines in main.c file:

- #define SSID "stwintest"
- #define PASSWORD "stwintest"

You can also edit the test_client_server.c file to set up a different address for the TCP echo server. By default, the ARM-Mbed echo server is used:

- #define REMOTE_IP_ADDR "52.215.34.155"
- #define REMOTE_PORT 7

RELATED LINKS

Appendix A Serial Terminal configuration using TeraTerm on page 10

3.3 BLE_SampleApp application

BLE_SampleApp provides an example of Bluetooth Low Energy configuration that enables the STWIN core system board to stream environmental sensor data; it is compatible with the ST BLE Sensor mobile app available for Android and iOS. BLE_SampleApp also shows how to correctly configure the STBC02 Battery Charger, providing a good example of how to manage power on/off routines for the STWIN board.

Following reset, the firmware performs the following initial tasks:

- configures HAL, clocks and buttons
- configures Battery Charger
- initializes the target platform:
- USB peripheral (for debugging)
  - LED1 and LED2
  - environmental sensors
  - initializes the Bluetooth Low Energy stack
- initializes the Bluetooth Low Energy services
- initializes timers
- starts the main loop:
  - LED management
  - BLE event management
  - environmental sensors data management

After you have downloaded the firmware, you can follow the procedure below to run the BLE SampleApp application:

Step 1. Once downloaded, the STWIN LED starts blinking.
This signals that the device is waiting for a connection via Bluetooth.

Step 2. Open the STBLESensor app on your Android or iOS smartphone and use it to connect to the STWIN core system board.

RELATED LINKS

Visit the ST website for information regarding ST BLE Sensor mobile app.

Appendix A Serial Terminal configuration using TeraTerm

Figure 9. **8N1, 115200 bauds, no HW flow control configuration**

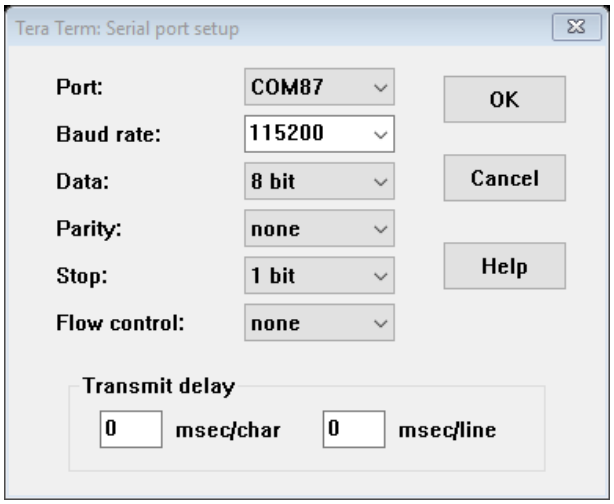
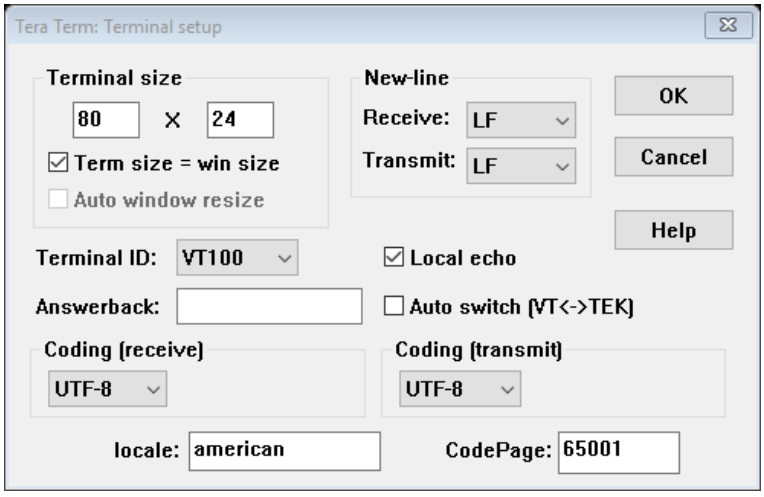


Figure 10. **Line endings to LF or CR-LF (Transmit) and LF (receive)**



Revision history

Table 1. **Document revision history**

Date	Version	Changes
05-Sep-2019	1	Initial release.
15-Nov-2019	2	Updated Introduction, Section 1 Overview, Section 2 How to program the board and Section 3.5 HS_Datalog application.
10-Feb-2020	3	Updated Introduction and Section 1 Overview. Added Section 3.3 UltrasoundFFT.
13-Jul-2020	4	Updated Introduction, Section 1 Overview, Section 2.1 How to program STWIN via STLINKV3MINI, Section 3 Sample applications, Section 3.4 HSDatalog application and Section 3.4.1 SD card considerations. Added Section 2.2 How to program STWIN using STM32CubeProgrammer "USB mode". Removed Section 3.2 OnboardMics application, Section 3.3 UltrasoundFFT and Section 3.7 MicArrayCoupon application.
13-Nov-2020	5	Updated Introduction, Section 1 Overview and Section 3 Sample applications. Removed Section 3.4 HSDatalog application. Added references to FP-SNS-DATALOG1.

Contents

List of figures