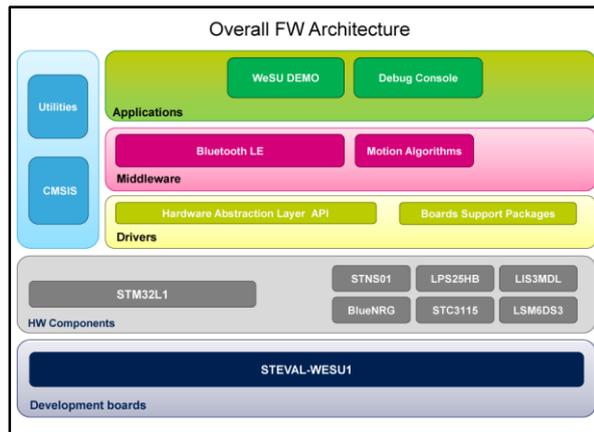


Bluetooth low energy, MEMS sensors and battery management firmware for wearable applications

Data brief



- Firmware package available in source code from ST website
- ST WeSU SDK and APPs available in source code from GitHub

Description

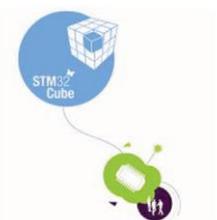
The STEVAL-WESU1 firmware provides a complete framework to build wearable applications, using inertial and environmental sensors, battery management, and Bluetooth low energy wireless communication. It is built on the STM32Cube™ for easy expansion and customization of the firmware for specific applications, such as integrating motion algorithms, or specific middleware components.

The firmware runs on the STM32L151 microcontroller and includes the drivers to interface with Bluetooth low energy (BlueNRG-MS), three MEMS sensor devices (LPS25HB, LSM6DS3, LIS3MDL), and two battery management devices (STNS01, STC3115). The BlueST protocol provides structured services and characteristics to expose the board functionalities and communicate with an Android or iOS smartphone via Bluetooth low energy wireless communication. Moreover, the ST WeSUApp provides specific demos for algorithms like AHRS, pedometer, free fall detection, as well as battery information and RF signal strength (RSSI).

The App also includes command line interface functionality via a debug console for smart device control by managing permanent and session settings registers of the STEVAL-WESU1.

Features

- Complete firmware architecture to build wearable applications using:
 - STM32L1 low power microcontroller
 - Inertial and environmental MEMS sensors
 - Bluetooth low energy (BLE) network processor
 - Several low power profiles
- Free comprehensive development firmware libraries and examples based on STM32Cube™
- Customized API to support specific hardware and firmware features of an autonomous wearable node: internal setting and functionalities can be controlled straight by the smartphone
- Middleware sensor fusion algorithm to combine the outputs from multiple motion sensors
- Based on BlueST protocol for data acquisition and logging; smartphones can be used to read sensor and algorithm data, as well as control the device via remote control settings and a debug console
- ST WeSU dedicated App, available from Apple Store™ and Google Play™ stores



1 Detailed description

Why use STM32Cube framework technology?

STM32Cube™ represents the STMicroelectronics initiative to make developers lives easier by reducing development effort, time and cost.

This STM32Cube framework contains:

- Firmware to build applications with the STM32 MCU to manage inertial and environmental sensors, power management devices and Bluetooth low energy wireless communication.
- Customized API to support the specific hardware functions of an autonomous wearable node, with a dedicated strategy to control its internal settings and functions.
- Firmware architecture totally with with STM32Cube rules and ready to accept any supplemental middleware.

How does this software embed STM32Cube?

The STEVAL-WESU1 FW extends the STM32Cube capabilities providing a board support package (BSP) customized for each subsystem (described below), and the proper middleware to facilitate the addition of further algorithms to the platform.

The overall architecture includes:

BSP

- **Component:** standard low level device drivers for LPS25HB, LSM6DS3, LIS3MDL, STC3115, including specific battery monitoring algorithm
- **WeSU:** (3 subsystems)
 - **BlueNRG:** APIs to control the BLE connectivity
 - **Platform:** APIs to control and configure Battery & PWR subsystem, plus generic functions directly accessible via the App (LED, push button, USB power)
 - **Sensors:** APIs to configure and control all the sensors

Middleware

- **STM32_BlueNRG**
- **AHRS algorithm**
- Ready for supplemental package integration: **motion algorithms**, etc.

User application

- Data streaming based on BlueST protocol
- Algorithms OTA/USB-DFU: over-the-air device firmware upgrade through BLE connectivity or USB 2.0 interface (*)
- Reset manager: application or OTA firmware boot selection (*)
- CLI: command line interface implementing a debug console
- Configuration interface: using EEPROM (permanent) and RAM (session) settings



(*) these packages are included in .bin files.

2 Revision history

Table 1: Document revision history

Date	Version	Changes
23-Feb-2016	1	Initial release.

IMPORTANT NOTICE – PLEASE 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 acknowledgement.

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. 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.

© 2016 STMicroelectronics – All rights reserved