AN5403

Application note

STEVAL-STRKT01 power management architecture description and configuration for optimized battery life

Introduction

The STEVAL-STRKT01 LoRa® IoT tracker is designed and optimized to implement the latest technologies in IoT tracker applications such as asset, people and animal tracking, as well as fleet management.

According to the selected use case, multiple features can be configured to satisfy application requirements and achieve the best performance in terms of power consumption and battery autonomy.

The STEVAL-STRKT01 building blocks reduce power consumption, which changes according to each sub-system hardware and firmware configuration.

The analysis performed and the examples shown can help users determine the best configuration for the STEVAL-STRKT01 in terms of sensor acquisition frequency, optimized sensor acquisition time, and any debug features necessary to achieve the longest possible battery life.

1 Acronyms and abbreviations

Acronym	Description
API	Application programming interface
GNSS	Global navigation satellite system
MCU	Microcontroller unit
LP	Low power
ULP	Ultra low power

2 STEVAL-STRKT01

2.1 Overview

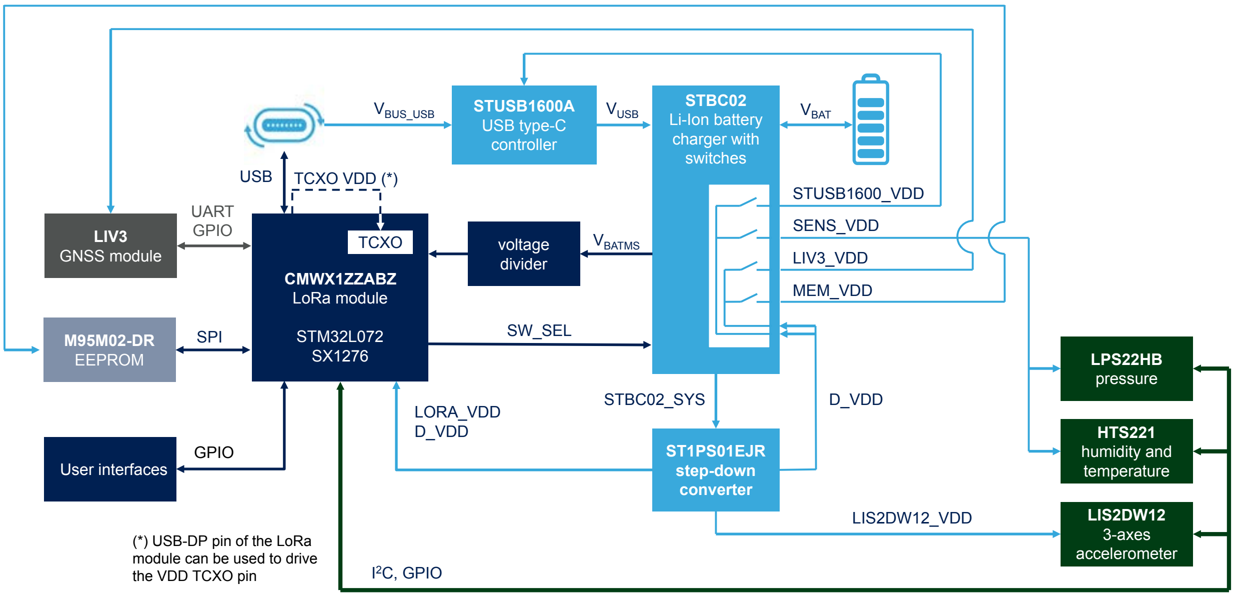
The STEVAL-STRKT01 is an optimized tracker solution over LoRaWAN network with simultaneous multi-constellation GNSS positioning and geofencing support.

It is battery operated with a versatile smart power management architecture covering different application profiles.

Some components and circuits supervise energy storage and manage energy supply modes, allowing the implementation of power management strategies.

The firmware acts on main power management main by disabling the power supply lines of subsystems not required by the application or by taking advantage of low power functions of the device.

Figure 1. STEVAL-STRKT01 block diagram and power management architecture



The main components involved in the power management strategies are the STUSB1600 (U500) USB Type-C controller, the STBC02 (U400) battery charger and the ST1PS01EJR (U401) step-down converter.

The STUSB1600 manages the USB Type-C attach/detach events and activates the 5 V supply path from the USB connector to the STBC02 battery charger.

The STBC02 power path management recharges the battery and supplies the system or allows the battery to power the device when the IN pin is not connected to a valid power source

The ST1PS01EJR is a nano-quiescent miniaturized synchronous step-down converter supplied by the STBC02 SYS pin. Its output, set to 3.3 V, is directly connected to the CMWX1ZZABZ LoRa module VDD pin and to the LIS2DW12 (U301) VDD pin.

The STBC02 hosts two SPDT load switches suitable to drive four supply lines:

- STUSB1600 VDD
- Sensors VDD (U303 and U304)
- Teseo-LIV3F (U200)
- M95M02-DR EEPROM (U600) memory

These switches can be activated by commands sent through a single wire connection from the STM32L0 MCU inside the CMWX1ZZABZ LoRa module.

Moreover, the integrated TCXO can be supplied by STM32L0 GPIO PA12, disabling it when not needed.

The TCXO driving is not enabled by default. As the GPIO number is limited, the STM32 PA12 is by default used for the USB communication. To implement this feature, a hardware modification is necessary (see Section 2.3 R106/R107 configuration for details).

2.2 Setup

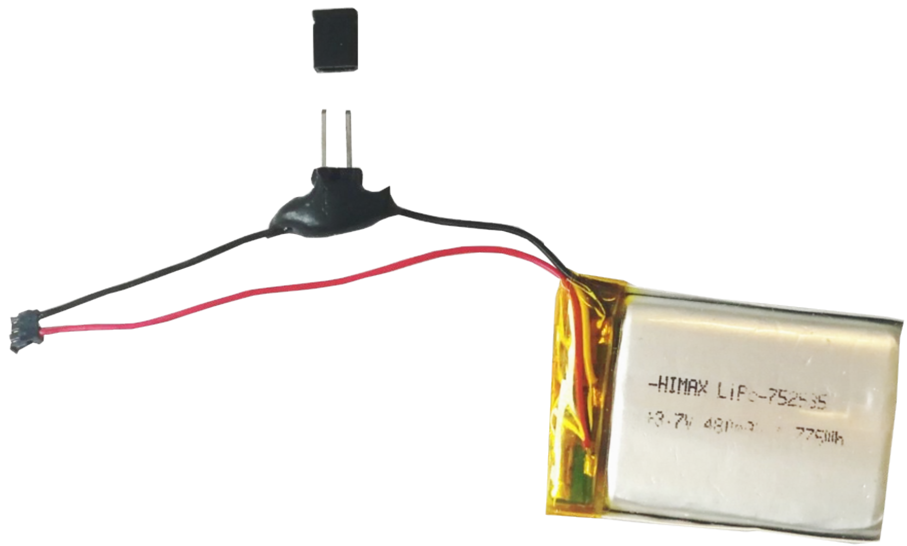
For energy consumption tests, follow the procedure below.

Step 1. Modify the battery to connect a jumper, a multimeter or a power analyzer.

Step 1a. To measure the STEVAL-STRKT01 current consumption, cut the battery ground wire and solder a two-pin header (e.g , PRPC002SAAN-RC) as shown in the following figure.

The header allows connecting a current measurement tool or a jumper.

Figure 2. STEVAL-STRKT01 battery modification for current measurement



Step 1b. Use a jumper to short circuit the two-pin header to recharge the battery.

The charger correctly determines the battery internal resistance.

Step 1c. Use a multimeter to measure the average current consumption, e. g. in low power modes.

Step 1d. Use a power analyzer to capture the current waveform of fast operations, such as radio transmissions.

Step 2. Configure the STEVAL-STRKT01 according to the test you want to perform (refer to Section 8 Power consumption tests), by removing some resistors and/or load a specific firmware

2.3 R106/R107 configuration

R106 and R107 resistors are used to supply the temperature controlled oscillator (TCXO) embedded in the Murata module (U100). The TCXO supply line is accessible through VDD_TCXO pin (U100 pin 48).

These resistors can be configured in a field configuration or a debug configuration.

Field configuration

In this configuration, the USB data functionality is not available and the power optimization is performed by removing R107, mounting R106 and driving VDD_TCXO pin through STM32L0 PA12 pin.

Debug configuration

In this configuration, the USB connection can be used, but without power consumption optimization. By default, R107 is mounted and R106 is not mounted, as shown in Figure 3. VDD_TCXO is directly connected to LORA_VDD and is always on.

The STM32L0 PA12 pin (U100 pin 1) is used for USB data functionality.

Figure 3. **STEVAL-STRKT01: R106/R107 configuration**

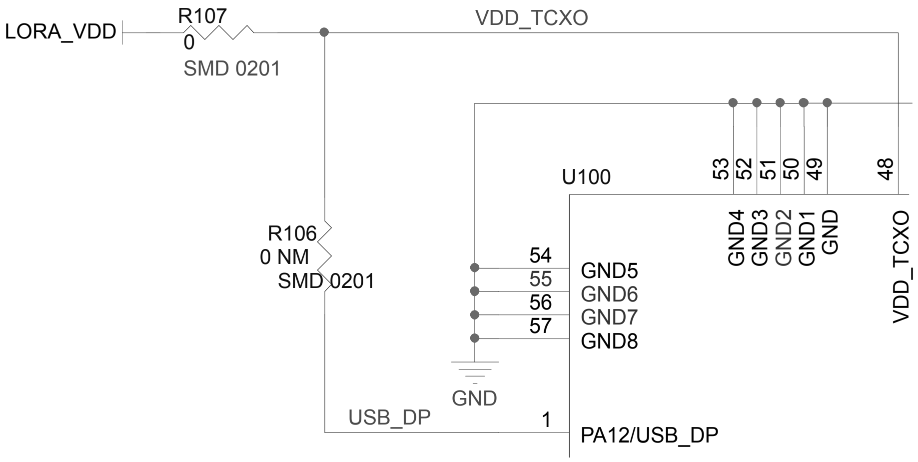


Figure 4. **STEVAL-STRKT01: R106 layout position (top view)**

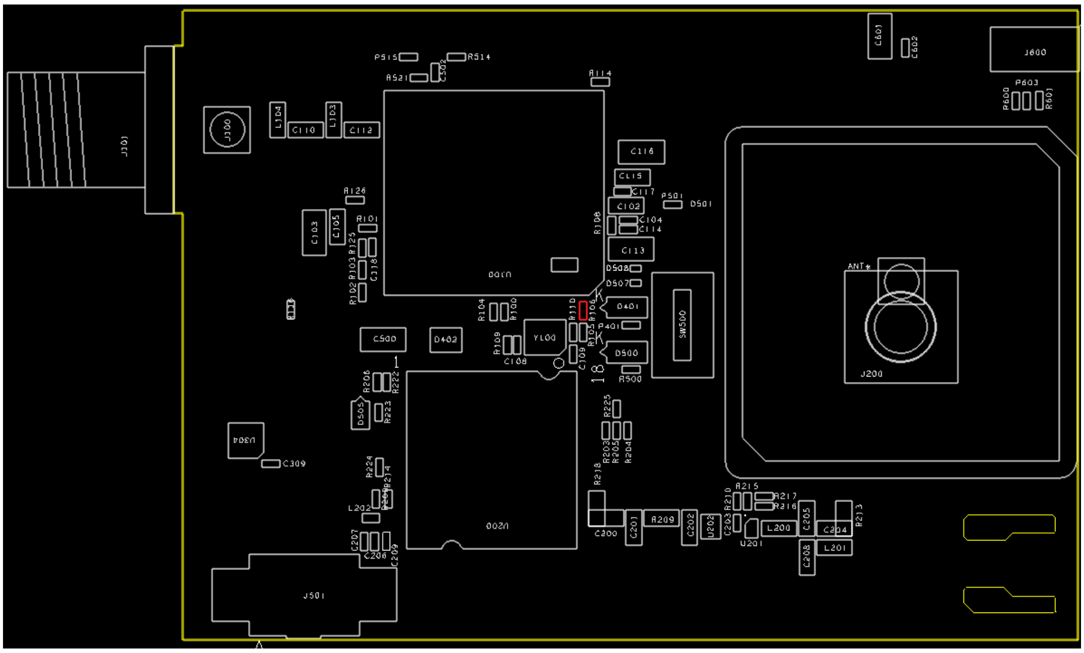
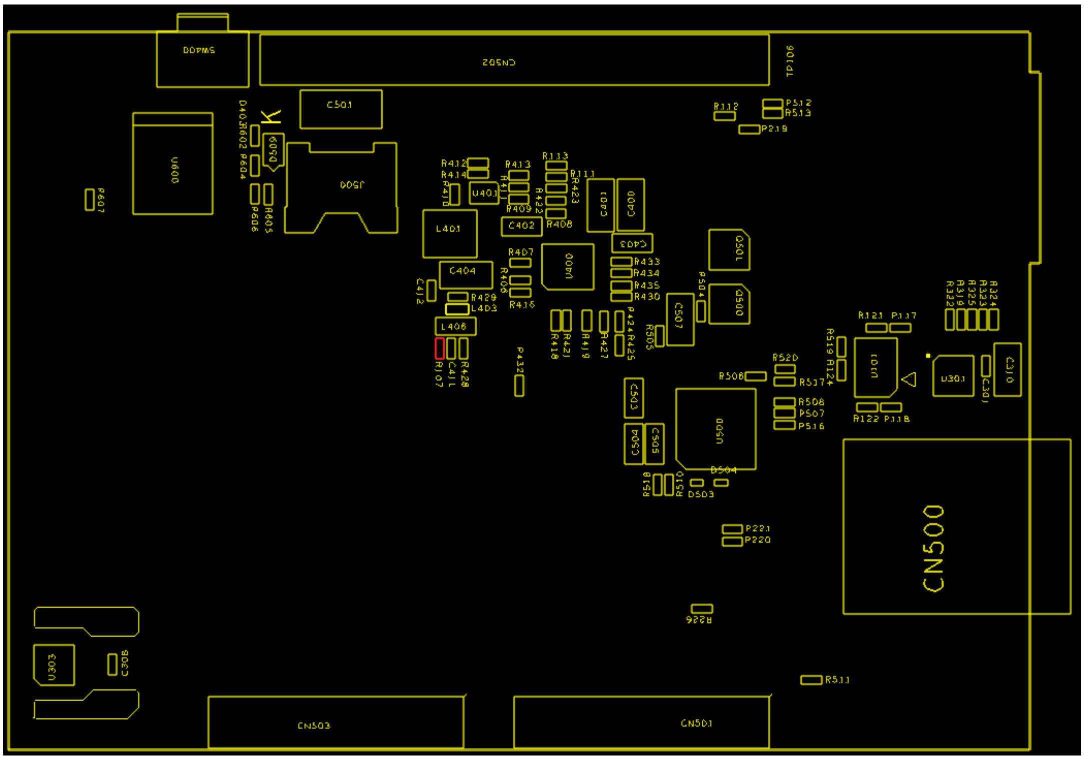


Figure 5. STEVAL-STRKT01: R107 layout position (bottom view)



2.4 LIS2DW12 SA0 pin configuration

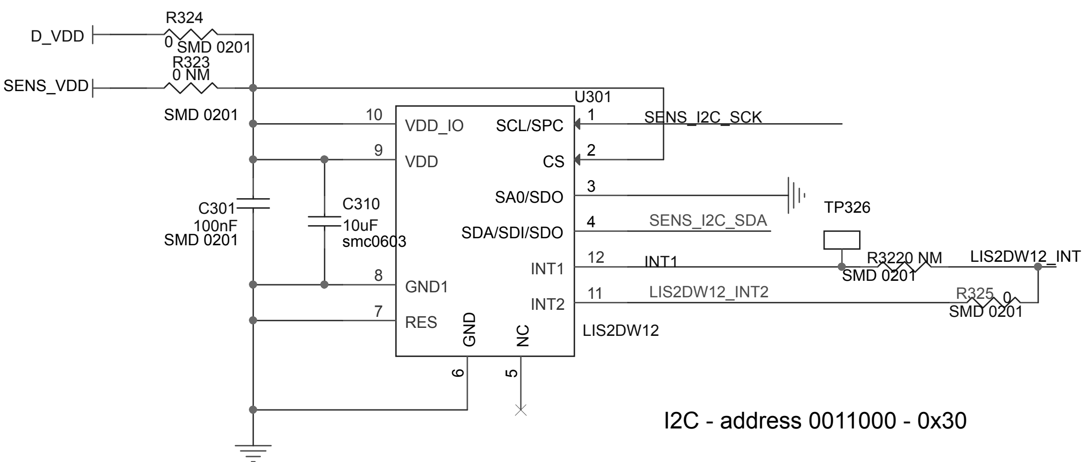
The LIS2DW12 (U301) 3-axis accelerometer is accessed through I²C interface.

Its slave address lowest bit, SA0, can be selected by connecting SA0/SDO (U301 pin 3) to GND or to VDD (refer to LIS2DW12 datasheet on www.st.com for further details).

The STEVAL-STRKT01 does not allow to change SA0 pin connection, even if the default configuration (connected to ground) causes a leakage of ~120 µA.

To optimize LIS2DW12 power consumption, connect SA0 pin to VDD.

Figure 6. STEVAL-STRKT01: LIS2DW12 schematic diagram



3 FP-ATR-LORA1 application firmware

The main function of the FP-ATR-LORA1 application firmware is to collect data coming from sensors, geo-position from GNSS and send them via LoRaWAN connectivity. It includes drivers for the LoRa radio, the Teseo-LIV3F GNSS module, the motion and environmental sensors, and the power management.

To ensure full system functionalities as well as long battery autonomy, the FP-ATR-LORA1 implements full run and low power profiles.

A state machine is implemented in the application firmware to manage the transitions among the various states.

3.1 FP-ATR-LORA1 state machine

The FP-ATR-LORA1 application firmware implements a state machine that can be used to develop, for example, asset tracking, fleet management and pet/child tracking applications.

The state machine is composed of three main states: Run, Low Power and Ultra Low Power. However, other temporary states (Read, Send, Check Acknowledgement, Retrieve Data, Prepare Low Power, Prepare Ultra Low Power) can be set to perform customized actions, such as reading sensors or sending data through the LoRaWAN network.

At system reset, the device remains running to collect data from sensors and GNSS and monitor the sensor values to detect if some thresholds are overshoot or a geofence alarm is reported by the GNSS.

When the accelerometer internal motion detection algorithm signals that the device is at rest, the device is put in low power mode (inactivity), disabling some sensors and/or power domains; the device can be woken up again by the accelerometer itself when it is moved (motion).

If the device remains in low power mode for a long time, a timer will trigger it to enter ultra-low power mode; in this state, the power consumption is reduced at the minimum. After entering the ultra-low power mode, the system wake ups periodically only to read sensors and send data. You can implement a check on the sensor data (acceleration, environmental data, geofence, etc.) to exit the ultra-low power mode and enter run mode (see `CheckSensorsData()` function in main.c).

Run: is the default state at system reset. All the devices are initialized and the application polls the environmental sensors, the accelerometer, the GNSS sensor and battery ADC converter for updated data.

Sensors and GNSS receiver are continuously monitored, to detect whether the sensor thresholds are overshoot or the GNSS sends geofence alerts. These events cause the system to go to Read And Send mode.

The accelerometer is also monitored to check:

- inactivity: no movement is detected for a certain amount of time
- wake-up: a movement is detected (the event can be triggered by briefly shaking the board)

Three timers are set to:

- switch to Read state at a user-configurable time interval
- switch to Send state
- switch to Ultra Low Power mode after a certain interval

All subsystems, sensors, GNSS and MEM_VDD are powered and the devices are running. The MCU is in run mode. STUSB1600_VDD is activated depending on the USB cable connection (for more details, see Table 3. STEVAL-STRKT01 power consumption).

Low Power: the MCU enters sleep mode to reduce power consumption and can be woken up by accelerometer interrupt events; the environmental sensors are disabled. If accelerometer wake-up event or timer event are detected, the system switches to Read state and, subsequently, to Send state. If accelerometer inactivity event is detected, the system switches to Ultra Low Power state. Before entering this state, a Prepare Low Power state is executed to disable unused power paths (no GNSS and no environmental sensors) and set a wakeup timer.

Ultra Low Power: the MCU enters Stop mode to reduce the power consumption at the minimum.

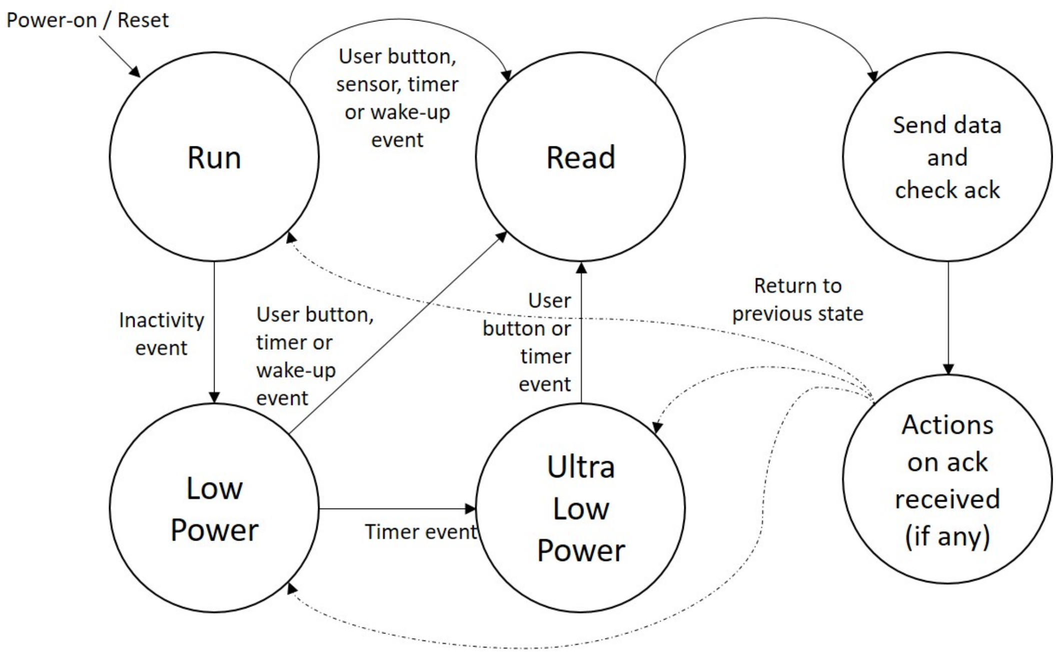
Environmental and accelerometer sensors are disabled and the MCU can be woken up only by the timer interrupt event. When this event is triggered, the system switches to Read state and, subsequently, to Send state.

Before entering Ultra Low Power state, a Prepare Ultra Low Power state is set to disable unused power paths: the sensors subsystem, SENS_VDD, the GNSS subsystem, LIV3_VDD and MEM_VDD, are not powered. The accelerometer is put in shutdown mode by software command. The MCU is in stop mode. STUSB1600_VDD is activated depending on the USB cable connection (for more details, see Table 3. STEVAL-STRKT01 power consumption).

Read: the application gathers data from environmental, accelerometer and GNSS. After the Read state, data are ready to be saved in the internal EEPROM and/or sent over the LoRaWAN network. The sensors are read and evaluated also in run mode. Anyway, this dedicated “temporary state” is implemented to have a coherent packet of data acquired from sensors at a specific timestamp.

Send: the application sends data to the LoRaWAN network packed in a single message. After the message is sent, the application performs the check on the acknowledgement (if this feature is enabled, i. e. `LORAWAN_DEFAULT_CONFIRM_MSG_STATE` define set to `LORAWAN_CONFIRMED_MSG`) and then the state machine switches back to the previous state.

Figure 7. **STEVAL-STRKT01: application state machine**



4 STEVAL-STRKT01 firmware update setup for energy profile configuration

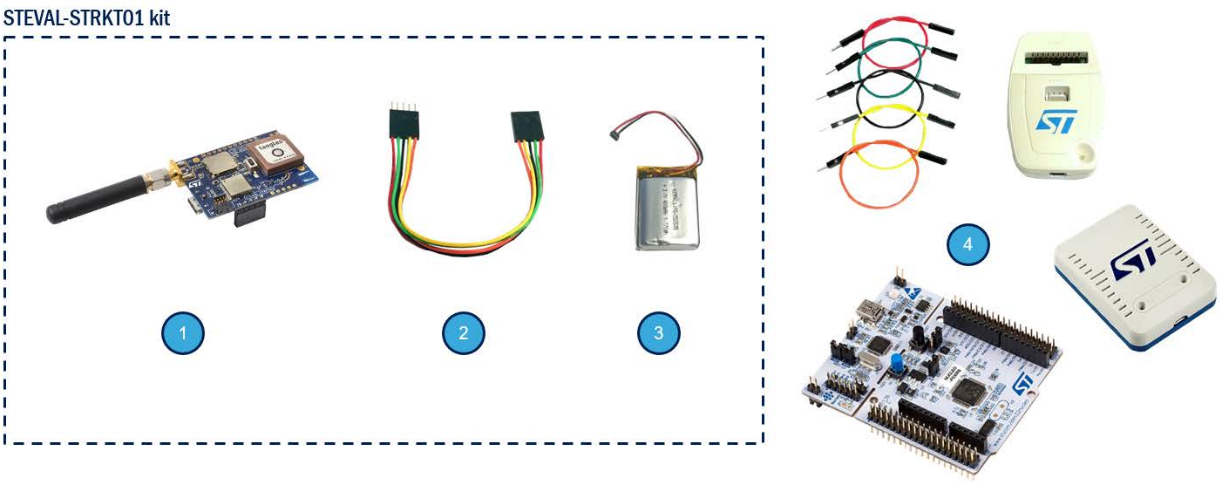
4.1 Hardware and software requirements

To modify the STEVAL-STRKT01 firmware for configuring different energy profiles, the following resources are needed (for further details refer to UM2541 and UM2487 on www.st.com):

- Hardware

Figure 8. Hardware resources required for STEVAL-STRKT01 firmware upgrade

1. STEVAL-STRKT01 evaluation board
2. Programming cable
3. Battery
4. One of the following in-circuit debugger/programmer: ST-LINK/V2, ST-LINK/V3 or ST-LINK/V2-1 (ST-LINK/V2-1 is part of any STM32 Nucleo-64 board, e.g. NUCLEO-F401RE)



- Software
  – FP-ATR-LORA1 (v2.1.1) STM32Cube function pack for IoT tracker node with LoRa connectivity, GNSS and sensors
  – Development tool-chain and Compiler: IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STR) or System Workbench for STM32 (SW4STM32)

4.2 How to load a new firmware on the STEVAL-STRKT01

Step 1. Download the FP-ATR-LORA1 STM32Cube function pack for STEVAL-STRKT01 IoT tracker node

Step 2. Customize the parameters in the firmware to run a specific test

Step 3. Compile the firmware

Step 4. Turn the board on and make sure that the battery is sufficiently charged to run the example
If you need to recharge the battery, see Section 2.2 Setup

Step 5. Download the binary file

Step 6. Switch the board off and restart it before performing power measurements
Refer to UM2541 and UM2487 on www.st.com for details.

4.3 Update via ST-LINK/V2 in-circuit debugger/programmer

Step 1. Connect the 5-pin flat cable (male side) to STEVAL-STRKT01 CN501 connector as per Table 1

Step 2. Connect the 5-pin flat cable (female side) to ST-LINK/V2 pins 1-5 as per Table 1

Step 3.

- **Step 3a.** Connect the battery and press SW400 button to power the board on (for 1.250 s at least)
  or
- **Step 3b.** Supply the STEVAL-STRKT01 via a Type-C USB cable (and a Type-C to Type-A adapter if needed)

Step 4. Connect the ST-LINK/V2 to a PC via a Type-A/mini B cable

Figure 9. **Hardware configuration for STEVAL-STRKT01 firmware update using an ST-LINK/V2 programmer**

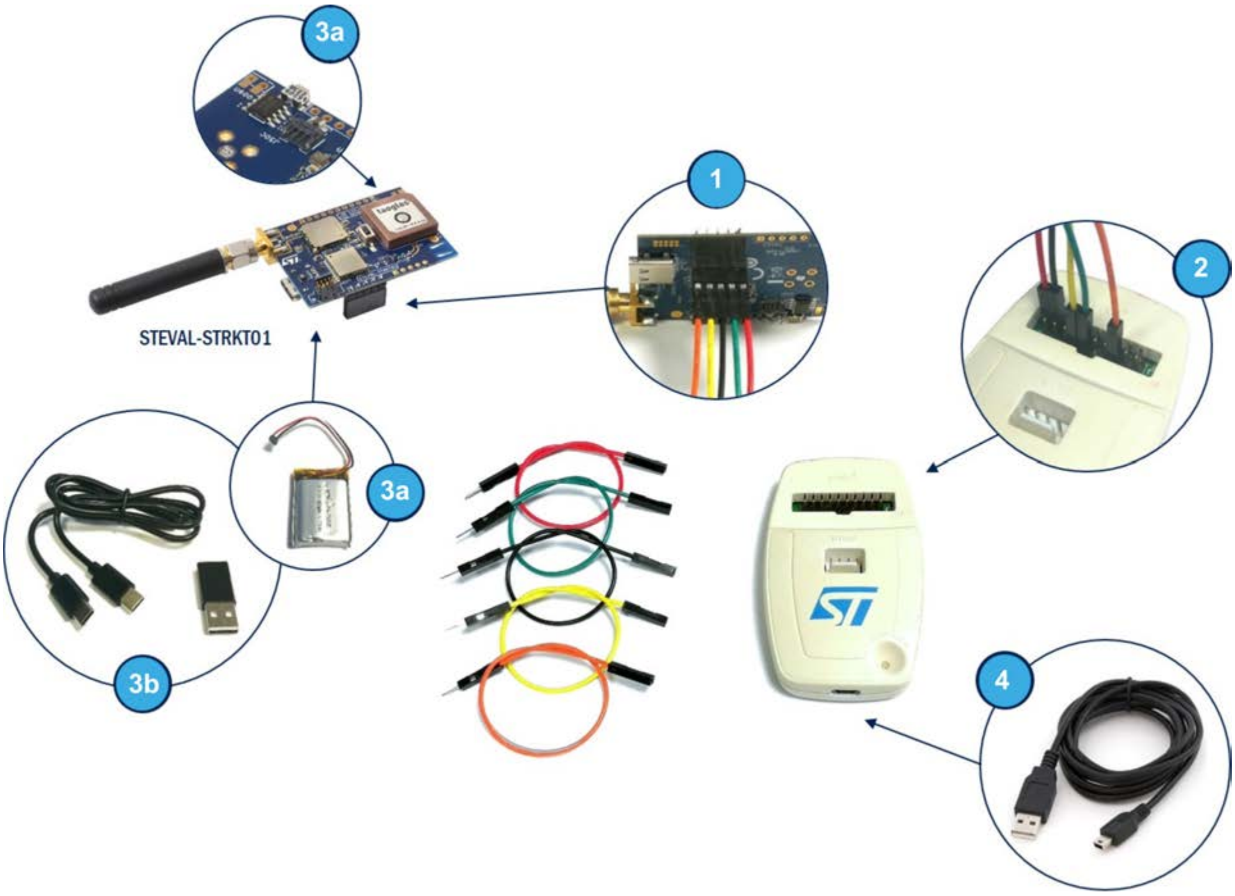
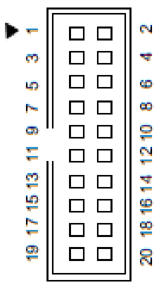


Table 1. **ST-LINK/V2 programmer and STEVAL-STRKT01 pinout**

ST-LINK/V2 connector (pin and label)		STEVAL-STRKT01 CN501 (pin and label)
	2, VAPP	5, D_VDD
	4, GND	3, GND
	7, TMS_SWDIO	2, SWD_SWDIO
	9, TCK_SWCLK	4, SWD_SWCLK
	15, NRST	1, nRESET

4.4 Update via ST-LINK/V3 in-circuit debugger/programmer

To perform the update via ST-LINK/V3 you first have to combine it with the adapter (MB1441 plus MB1440, shown in Figure 10. Hardware configuration for STEVAL-STRKT01 firmware update using an ST-LINK/V3 programmer).

Step 1. Connect the 5-pin flat cable (male side) to STEVAL-STRKT01 CN501 connector

Step 2. Connect the 5-pin flat cable (female side) to MB1440 CN6 connector pins 1-5 and leave pin 6 unconnected

Step 3.

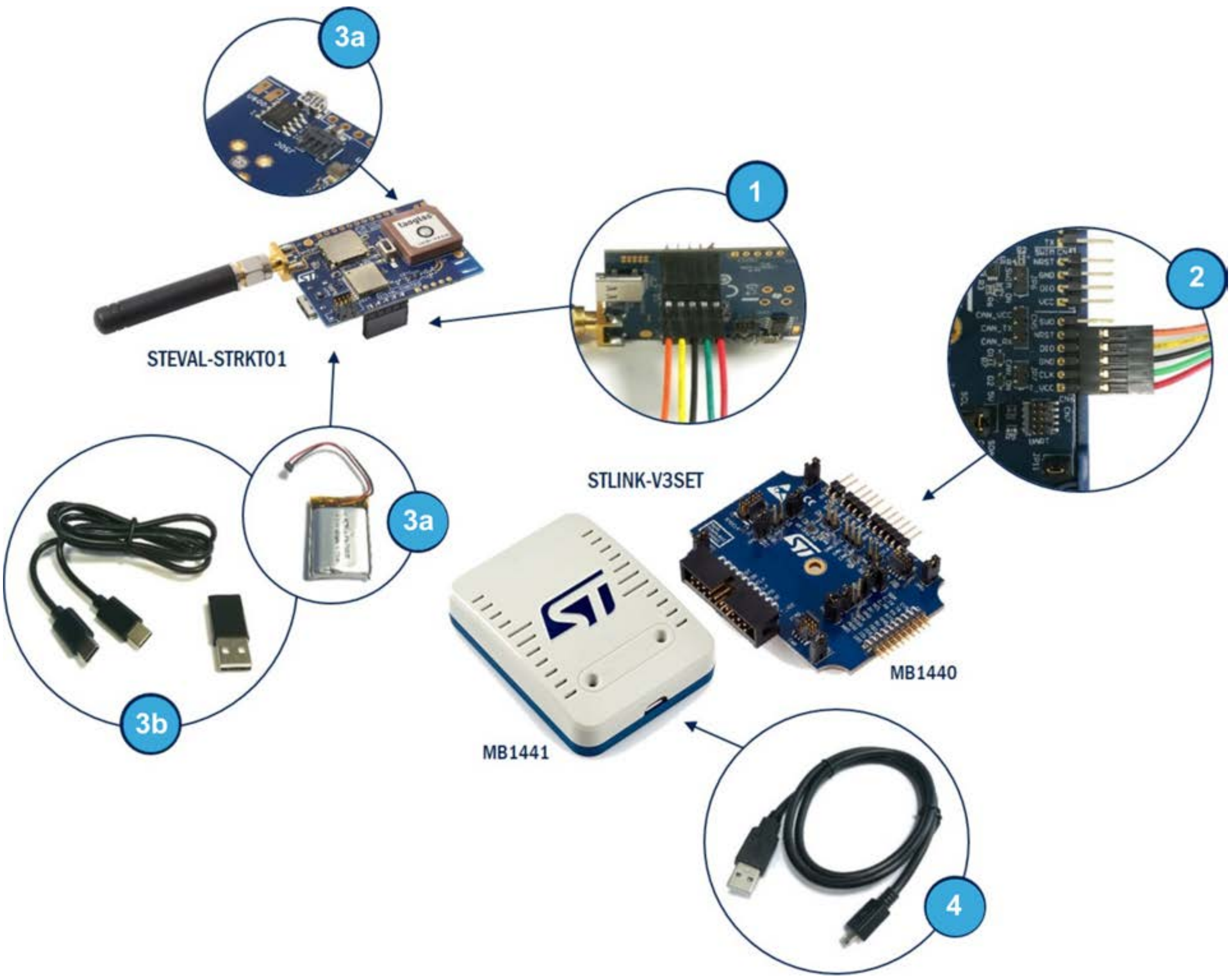
- **Step 3a.** Connect the battery and press SW400 button to power the board on (for 1.250 s at least)

  or

- **Step 3b.** Supply the STEVAL-STRKT01 via a Type-C USB cable (and a Type-C to Type-A adapter if needed)

Step 4. Connect the ST-LINK/V3 to a PC via a Type-A/micro B cable

Figure 10. Hardware configuration for STEVAL-STRKT01 firmware update using an ST-LINK/V3 programmer



4.5 Update via STM32 Nucleo-64 on-board ST-LINK programmer

Step 1. Connect the 5-pin flat cable (male side) to STEVAL-STRKT01 CN501 connector

Step 2. Connect the 5-pin flat cable (female side) to the STM32 Nucleo SWD connector pins 1-5, leave pin 6 unconnected and remove CN2 jumpers

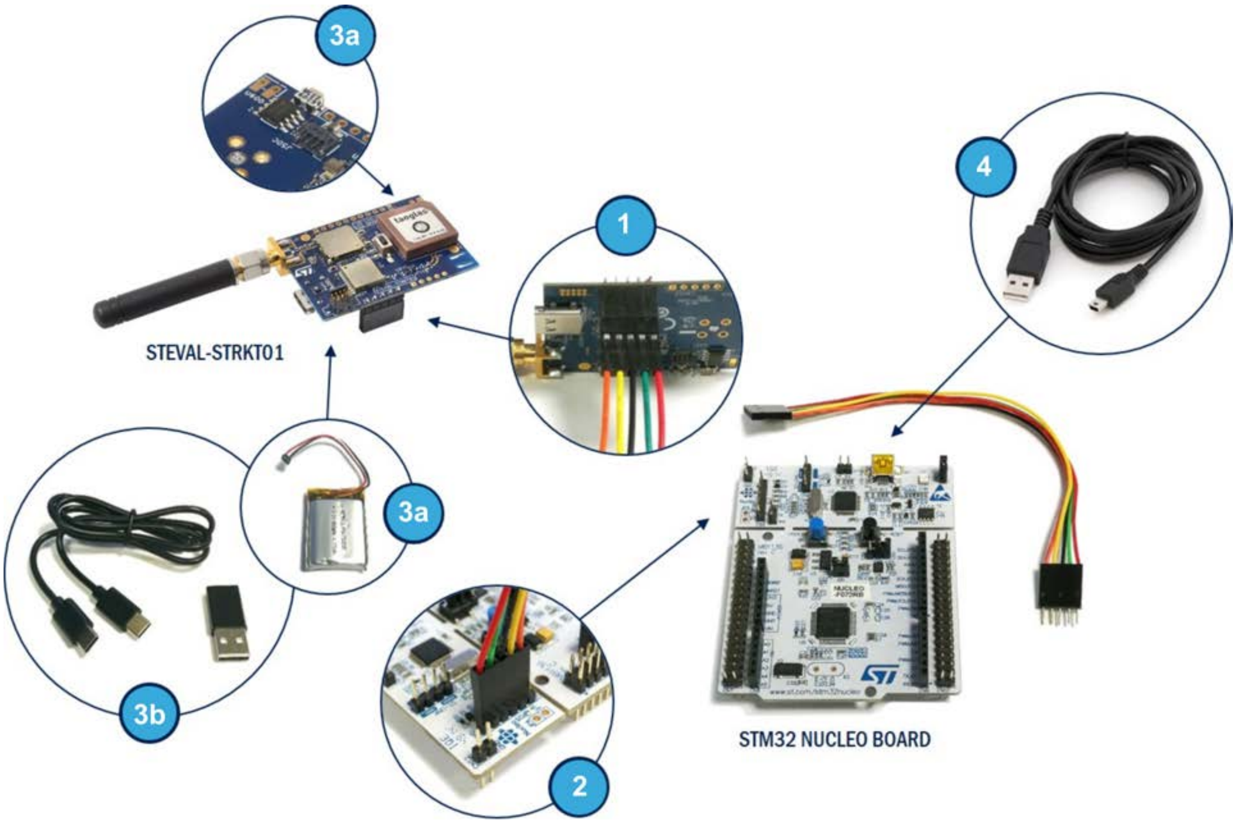
Step 3.

Step 3a. Connect the battery and press SW400 button to power the board on (for 1.250 s at least) or

Step 3b. Supply the STEVAL-STRKT01 via a Type-C USB cable (and a Type-C to Type-A adapter if needed)

Step 4. Connect the STM32 Nucleo to a PC via a Type-A/mini B cable

Figure 11. Hardware configuration for STEVAL-STRKT01 firmware update using STM32 Nucleo-64 on-board ST-LINK programmer



5 Firmware configuration

To perform power consumption optimization tests, you have to apply specific configurations to the firmware.

Some settings can be changed from a #define in the firmware, others can be changed both at runtime by USB commands (if USB is available) and by the corresponding #define in the firmware.

5.1 Enabling/disabling the USB

The firmware can be configured to enable the USB communication (the hardware must be configured accordingly) or disable it. Both configurations are done by acting on the `USB_ENABLED` define in *hw_conf.h* file.

By uncommenting the `USB_ENABLED` define, the firmware is configured to use the USB interface and resistors R106/R107 must be configured accordingly, as described in Section 2.3 R106/R107 configuration (debug configuration).

By commenting the `USB_ENABLED` define, the firmware is configured to disable the USB interface and resistors R106/R107 configuration generally used is as described in Section 2.3 R106/R107 configuration (field configuration).

5.2 Enabling/disabling TCXO VDD management

The firmware can be configured to manage the TCXO power, to activate it only during LoRa transmissions and save power when not needed. It can be managed by using the same pin for USB communication; so this configuration is mutually exclusive with USB enabling.

In *hw_conf.h*, the `USE_TCXO_VDD` define can be used to enable/disable the TCXO VDD management. The hardware must be configured accordingly.

By setting the `USE_TCXO_VDD` define to 1, the firmware is configured to manage the TCXO VDD by the STM32 GPIO PA12 and resistors R106/R107 must be configured as described in Section 2.3 R106/R107 configuration (field configuration). The USB cannot be used in this case.

By setting the `USE_TCXO_VDD` define to 0, the firmware does not manage the TCXO VDD and resistors R106/R107 must be configured as described in Section 2.3 R106/R107 configuration (debug configuration). USB can be enabled.

5.3 Enabling/disabling debug features

The firmware can be configured to enable or disable the debug features, such as the SWD debug interface and the trace messages, in the *hw_conf.h*.

By uncommenting the `DEBUG` define, the firmware is configured to use the debug interface, i.e. SWD for in-circuit debugger connection.

By commenting the `DEBUG` define, the firmware is configured to disable the debug interface helping to save power.

5.4 Enabling/disabling memory

The firmware can be configured to enable or disable the features managing the M95M02-DR EEPROM memory by using the `MEMORY_ENABLED` define in the *hw_conf.h* file.

By uncommenting the `MEMORY_ENABLED` define, the firmware is configured to use the M95M02-DR EEPROM memory.

By commenting the `MEMORY_ENABLED` define, the firmware is configured to disable the M95M02-DR EEPROM memory usage.

5.5 GNSS power management

The firmware can be configured to put the Teseo-LIV3F module in power saving mode by:

- shutting down the module, switching its power line off
- sending a standby software command to put it in standby mode

In the *hw_conf.h* file, `GNSS_STANDBY_WAKEUP` define is used to enable/disable the GNSS standby command.

If GNSS_STANDBY_WAKEUP is set to 1, the GNSS is put in standby mode, through a command via serial interface, when it is not in use. This helps to speed the time up to the first fix at the next GNSS wakeup.

If GNSS_STANDBY_WAKEUP is set to 0, the GNSS VDD supply line is powered off by disabling the associated STBC02 switch through a command on the single wire interface, when the GNSS is not in use. In this case the time for the first fix will be longer.

5.6 Read timer

The firmware can be configured to periodically trigger data acquisition and sending.

If APPLICATION_READ_TIMER = 1, the setting is enabled.

If APPLICATION_READ_TIMER = 0, the setting is disabled.

5.7 Low power settings

The accelerometer inactivity event can make the STEVAL-STRKT01 board enter low power mode. The same result can be obtained with a timer.

The accelerometer can be configured to rise an interrupt after a period of inactivity, i. e. no motion detected. This event triggers the state machine change from full run to low power.

When the system is in low power, a timer is configured to switch to ultra-low power mode after a configurable time interval (see SLEEP_TIMER_INTERVAL define in the firmware library).

Low power mode enabled

Use the following commands to enable the low power mode triggering:

- !lpsensorevent-1 to enter low power mode on an accelerometer inactivity interrupt
- !lpsleeptimer-1 to go from low power to ultra-low power mode after a certain time interval

If USB interface is not available, change the USE_EEPROM_SETTINGS to 0 and the corresponding values in the firmware:

- LOW_POWER_ON_SENSOR_EVENT_DEF = 1 to enter low power mode on an accelerometer inactivity interrupt
- LOW_POWER_ON_SLEEP_TIMER_DEF = 1 to go from low power to ultra-low power mode after a certain time interval

Low power mode disabled

Use the following commands to disable the low power mode triggering:

- !lpsensorevent-0 to disable accelerometer inactivity triggering
- !lpsleeptimer-0 to disable ultra-low power triggering through the timer

If the USB interface is not available, change the USE_EEPROM_SETTINGS to 0 and the corresponding values in the firmware:

- LOW_POWER_ON_SENSOR_EVENT_DEF = 0 to enter low power mode on an accelerometer inactivity interrupt
- LOW_POWER_ON_SLEEP_TIMER_DEF = 0 to go from low power to ultra-low power after a certain time interval

6 Power management key points

The STEVAL-STRKT01 board offers a wide flexibility that allows the evaluation of different features and use cases.

The board power consumption is closely related to the chosen configuration. For this reason, to evaluate energy consumption, it is necessary to take into account the key points impacting power consumption as well as the identified use case:

- state machine: the number of times the “run” and “send” states are executed impacts power consumption
- power management device: STBC02 allows switching subsystems off, disabling functionalities when they are not used
- Teseo-LIV3F power consumption in standby mode is around 14 μA. This power consumption can be reduced to zero by switching Teseo-LIV3F VDD off . However, in the first case, the GNSS does not need to resynchronize its RTC at every wakeup, thus the operating time-to-achieve-the-first-fix (TTFF) is reduced.
- VDD TCXO Murata module: according to the STEVAL-STRKT01 default configuration, the VDD TCXO is always on. However, the LoRa module USB-DP pin can be used to drive the VDD TCXO pin, drastically reducing power consumption to ultra-low power state, by acting on resistors R106 and R107. The drawback is the loss of USB data communication feature.

6.1 Power consumption optimization

By default, the STEVAL-STRKT01 board starts in RUN state but can switch to different states according to the application needs.

For power consumption optimization the MCU, the sensors and the radio module should be active only when needed, to retrieve data from sensors and send them via LoRa. In the remaining time, everything must be put at the lowest current consumption state.

To select the right configuration for the board, you need to know (referring to Q1, Q2, Q3a and Q3b in Figure 12. Low power configuration selection scheme):

1. if the continuous monitoring for environmental sensors is necessary
   - if yes → only full run can be selected
   - if no → select the low power mode and go ahead
2. if the debug interface is needed and if any command should be sent via USB
   - if yes → you need USB connection
   - if no → USB GPIOs can be disconnected and power consumption is optimized: one of the USB GPIOs can be used to drive the VDD TCXO, reducing the power consumption in low power and ultra-low power modes
3. when you need the USB functionality, check if the sending interval is less than 4 hours
   a. ◦ if yes → GNSS time-to-first-fix (TTFF) can be optimized. If GNSS RTC is always powered, there is no need to re-synchronize the RTC at the next wakeup, performing a hot start; this is possible if ephemeris are up-to-date (i.e. max 4 hours old). During low power, the GNSS is put in standby via a software command (refer to Teseo-LIV3F documentation on www.st.com for further details). In these conditions, the TTFF can be optimized down to 5 seconds (2 seconds in theory)
      ◦ if no → GNSS TTFF cannot be optimized. If the GNSS remains in standby for more than 4 hours, the satellites ephemeris will be invalidated, so it is useless to put the GNSS in standby. To save power, the best choice is to remove the power from GNSS. In these conditions, the TTFF is around 40 seconds, as the Teseo-LIV3F performs a cold start
   b. when you do not need the USB functionality, check if the sending frequency is less than 4 hours
      ◦ if yes → GNSS TTFF can be optimized (GNSS RTC is always on, see 3a)
      ◦ if no → GNSS TTFF cannot be optimized

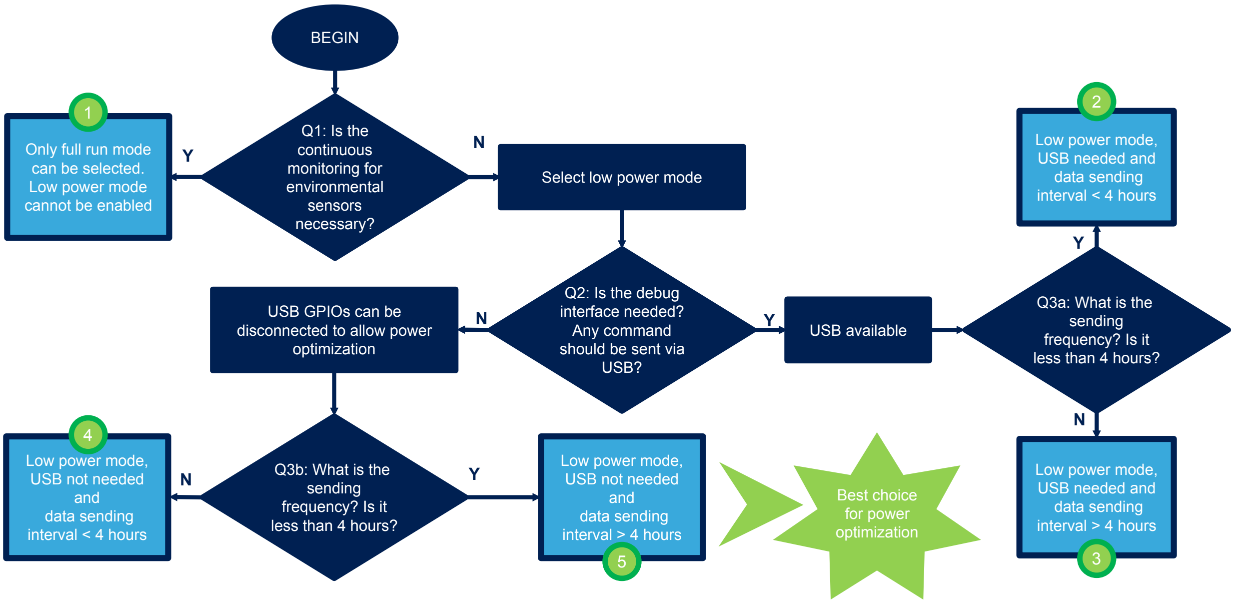


Figure 12. Low power configuration selection scheme

7 Low power configurations

To implement the configuration described in Figure 12. Low power configuration selection scheme, you need to perform some hardware and/or firmware configurations.

Important: Recompile the code to apply the firmware modifications.

The following table lists the current consumption related to low power configurations.

Table 2. Current consumption in different low power configurations

Configuration		Functionality			
		Low power mode	USB availability/ TCXO VDD management	GNSS in low power mode	Current consumption
1	Sensors continuous monitoring (no low power mode)	OFF	Don't care	Don't care	58 mA
2	Low power mode USB needed (for debug purpose or command availability) Data sending interval < 4 hours [(1)]	ON	USB availability	Standby	1.4 mA
3	Low power mode USB needed (for debug purpose or command availability) Data sending interval >4 hours[(1)]	ON	USB availability	OFF	1.4 mA[(2)]
4	Low power mode USB not needed Data sending interval <4 hours[(1)]	ON	TCXO VDD management	Standby	169 µA
5	Low power mode USB not needed Data sending interval >4 hours [(1)]	ON	TCXO VDD management	OFF	155 µA

1. *This duration is the max. validity time for ephemeris that allows having an advantage in terms of energy consumption by sending the standby command to the Teseo-LIV3F module instead of switching its power supply line off.*
2. *Switching the GNSS power supply line off reduces the current consumption by about 14 µA.*

7.1 Configuration 1: sensor continuous monitoring

As described in Section 5.7 Low power settings the low power mode must be disabled by acting on the following firmware commands:

- `!lpsensorevent-0`
- `!lpsleeptimer-0`

In this configuration, the USB/TCXO functionality can be selected according to the user needs (refer to Section 2.3 R106/R107 configuration for the consequent R106, R107 choice). To benefit from the USB functionality, the TCXO is not managed, causing 1.4 mA of extra current consumption.

7.2 Configuration 2: low power mode, USB used, data sending interval less than 4 hours

To implement this case, follow the procedure below.

Step 1. Activate the low power mode functionality. This can be done as described in Section 5.7 Low power settings, by acting on the following firmware commands.
- `!lpsensorevent-1`
- `!lpsleeptimer-1`

Step 2. Use the USB functionality after checking the hardware configuration (R107 mounted, R106 not mounted), as described in Section 5.1 Enabling/disabling the USB.

Step 3. Activate the USB feature, by uncommenting the `USB_ENABLED` define, in the *hw_conf.h* file.

Step 4. As the sending interval is less than 4 hours, set the `GNSS_STANDBY_WAKEUP` to 1 in the *hw_conf.h* file.
When the GNSS is not used, it is set to standby mode by a command via serial interface to speed TTFF up at next GNSS wakeup.

7.3 Configuration 3: low power mode, USB used, data sending interval more than 4 hours

To implement this case, follow the procedure below.

Step 1. Follow steps 1-2-3 of use case 2.

Step 2. As the sending interval is more than 4 hours, set the `GNSS_STANDBY_WAKEUP` to 0 in the *hw_conf.h* file.
When the GNSS is not used, its VDD supply line is powered off by disabling the associated STBC02 switch through a command on the single wire interface.

7.4 Configuration 4: low power mode, USB is not used, data sending interval less than 4 hours

To implement this case, follow the procedure below.

Step 1. Follow step 1 of use case 2.

Step 2. Disable the USB functionality after checking the hardware configuration (R107 not mounted, R106 mounted), as described in Section 5.1 Enabling/disabling the USB.

Step 3. Disable the USB feature, by commenting the `USB_ENABLED` define, in the *hw_conf.h* file.

Step 4. As the sending interval is less than 4 hours, set the `GNSS_STANDBY_WAKEUP` to 1 in the *hw_conf.h* file.
When the GNSS is not used, it is set to standby mode by a command via serial interface to speed TTFF up at next GNSS wakeup.

7.5 Configuration 5: low power mode, USB is not used, data sending interval more than 4 hours

To implement this case, follow the procedure below.

Step 1. Follow steps 1-2-3 of use case 4.

Step 2. As the sending interval is more than 4 hours, set the `GNSS_STANDBY_WAKEUP` to 0 in the *hw_conf.h* file.
When the GNSS is not used, its VDD supply line is powered off by disabling the associated STBC02 switch through a command on the single wire interface.

As described in Table 2. Current consumption in different low power configurations, this last case assures the best performance in terms of current consumption.

8 Power consumption tests

8.1 Battery charge

Test conditions:

- charger settings: I_{SET}=100 mA, I_{PRE}=10 mA
- firmware configuration: full run
- battery is fully discharged: test starts when Type-C connector is plugged in

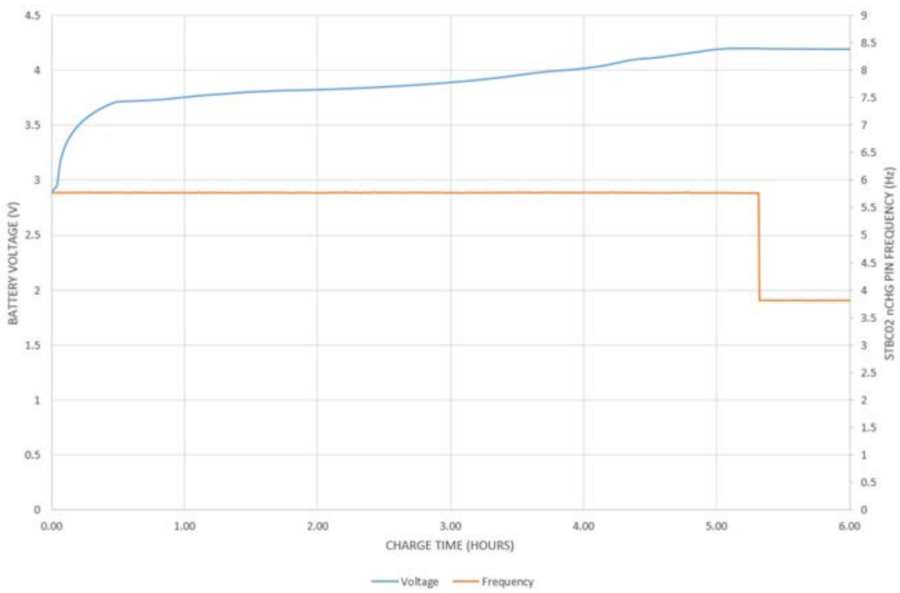
Background

The STEVAL-STRKT01 board embeds the STBC02 battery charger allowing single-cell Li-Ion and Li-Poly battery chemistry to be charged up to a 4.45 V using a CC-CV charging algorithm.

The charging cycle starts when a valid input voltage source is detected and signaled by the CHG pin toggling (6.2 Hz) from a high impedance state to a low logic level. If the battery is totally discharged, the STBC02 charger enters the pre-charge phase and starts charging in a constant current mode with the pre-charge current (I_{PRE}) set. By contrast, as soon as the battery voltage reaches the V_{PRE} threshold, the constant current fast charge phase starts and the relevant charging current increases to the I_{FAST} level.

The constant-current fast charge phase lasts until the battery voltage is lower than V_{FLOAT}. Consequently, the charging algorithm switches to a constant voltage (CV) mode. During the CV mode, the battery voltage is regulated to V_{FLOAT} and the charging current starts decreasing over time. As soon as it goes below I_{END}, the charging process is considered to be completed (EOC, end-of-charge) and the relevant status is shown as a 4.1 Hz toggling signal on the CHG pin.

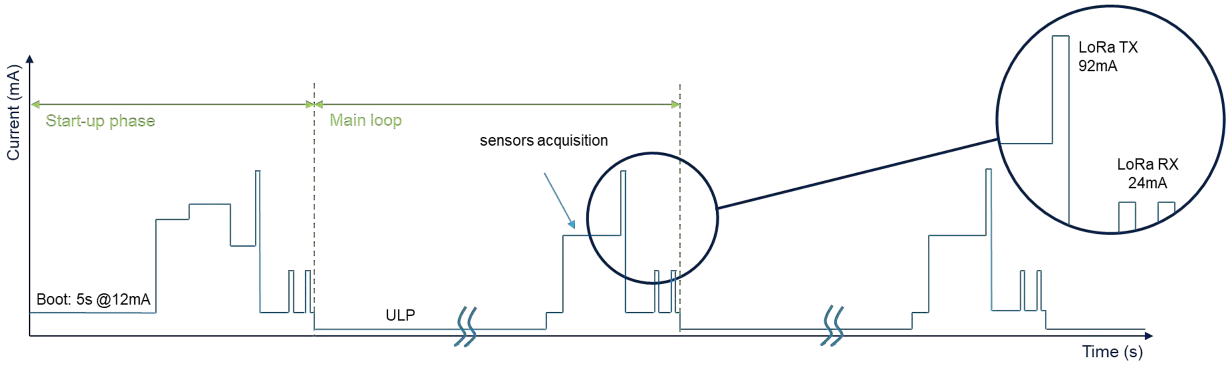
Figure 13. **STEVAL-STRKT01 battery charge test (battery capacity = 480 mAh)**



8.2 Battery discharge

8.2.1 Overview

Figure 14. **STEVAL-STRKT01 typical current consumption waveform**



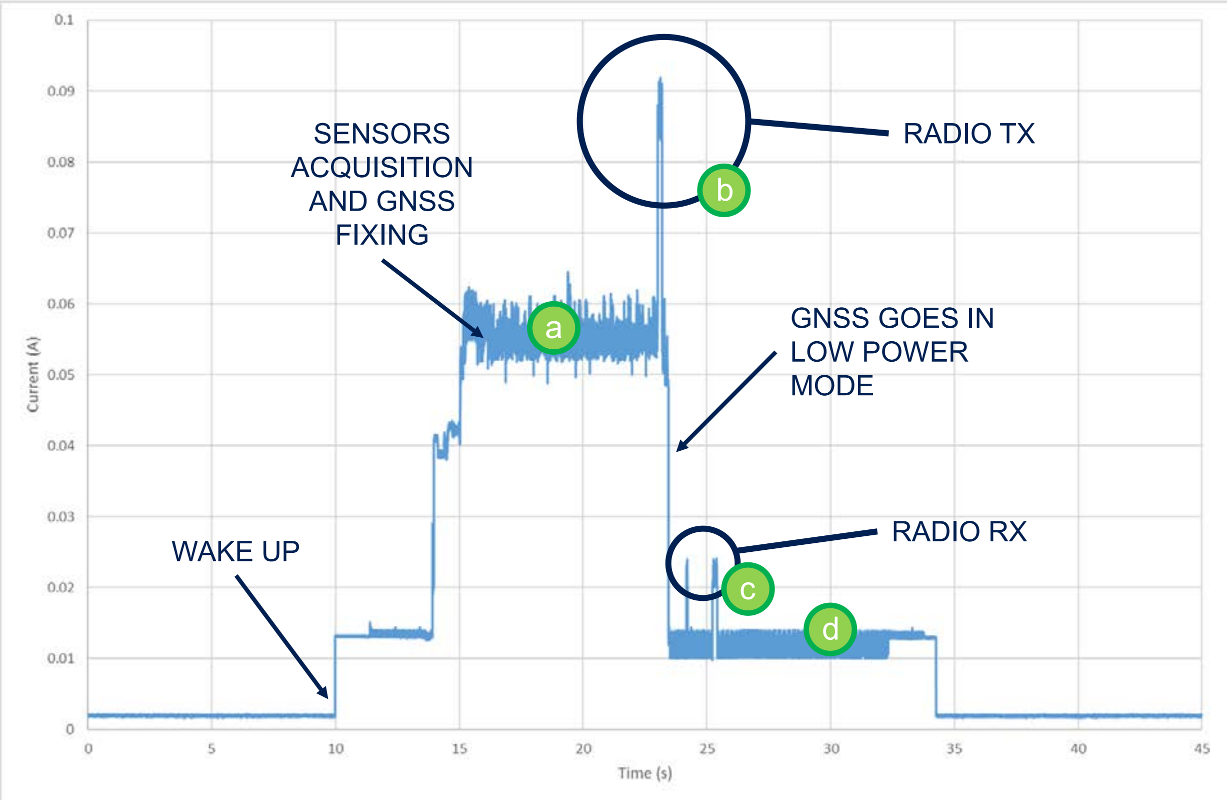
At system startup, some initialization operations are performed. The microcontroller configures all the subsystems, it carries out a first measurement cycle and registers the node in the network. Thus, according to the allowed customization, the MCU typically drives the system in low power state to reduce the energy consumption and subsequently it moves to the ultra-low power state if an accelerometer inactivity event is detected.

A wakeup timer periodically triggers the system in ultra-low power state moving it from ultra-low power state to RUN state followed by the SEND state. Therefore, the system goes back to ultra-low power state.

This main loop is repeated until the energy stored in the battery is enough to support it.

Figure 15. STEVAL-STRKT01 wakeup cycle waveform

a. $I_{AVERAGE}$ = 56 mA
b. I_{MAX} = 92 mA, $I_{AVERAGE}$ = 88 mA, t_{TX} = 175 ms
c. I_{MAX} = 24 mA, $I_{AVERAGE}$ = 22 mA, t_{RX1} = 50 ms, t_{RX2} = 197 ms
d. $I_{AVERAGE}$ = 12 mA



Differently from the wakeup cycle, the energy of the main loop is closely related to the number and type of performed actions, as the energy efficiency of an ultra-low power system is based on the chosen components as well as on the implementation of operating strategies able to guarantee the expected functions with the lowest energy impact.

System states (run, data acquisition, data transmission and low/ultra-low power) have an energy impact that depends on the average current and the time duration.

The average current consumption during each phase is shown in the following table (for low power modes, transmissions are not taken into account).

Table 3. STEVAL-STRKT01 power consumption

Mode	MCU	Radio	GNSS	Sensors	TCXO VDD	Total current
RUN	RUN	Idle	ON	ON	ON(1)	56 mA
Transmission	RUN	TX	ON	ON	ON (1)	92 mA
Low power	SLEEP mode	Idle	OFF	ON	ON (1)	4.6 mA
Ultra low power 1	STOP mode	Idle	OFF	OFF	ON (1)	1.4 mA
Ultra low power 2	STOP mode	Idle	Stand-by	OFF	OFF(2)	169 µA
Ultra low power 3	STOP mode	Idle	OFF	OFF	OFF (2)	155 µA
Ultra low power 4(3)	STOP mode	Idle	OFF	OFF	OFF (2)	39 µA

1. *Configuration described in Section 2.3 R106/R107 configuration (debug configuration).*

2. *Configuration described in Section 2.3 R106/R107 configuration (field configuration).*
3. *Configuration described in Section 2.4 LIS2DW12 SA0 pin configuration.*

On the basis of the consumption data shown above, some cases have been identified and tests performed, whose results are shown hereafter.

8.2.2 Ultra-low power test

The applied hypotheses match well with anti-tamper applications: the STEVAL-STRKT01 board is always sleepy and it wakes up only on an external trigger.

Test conditions

We assume that the battery is fully charged (V_{BAT} = 4.2V). The STEVAL-STRKT01 board is configured to achieve the lowest power consumption, supposing that:

- MCU is in STOP mode
- Radio is in IDLE state
- GNSS feeding line is off
- Sensors feeding line is off
- VDD_TCXO pin is managed by the microcontroller

The test starts when the Type-C connector is unplugged.

Results and comments

The test aims at assessing the battery life that allows the system to react to that asynchronous event.

The estimated battery life in the stated conditions is 129 days.

Note: It can be extended to 512 days by connecting SA0 to V_{DD} (as a suggestion for your implementation).

8.2.3 Full run test

The applied hypotheses match well with pet tracking applications.

Test conditions

We assume that the battery is fully charged (V_{BAT} = 4.2 V). The STEVAL-STRKT01 board is configured in full run (i. e. to continuously monitor sensors and GNSS receiver) to detect whether the sensor thresholds are overshoot or the GNSS sends geofence alerts and sends data to the LoRaWAN network (see Figure 12. Low power configuration selection scheme).

The test starts when the Type-C connector is unplugged.

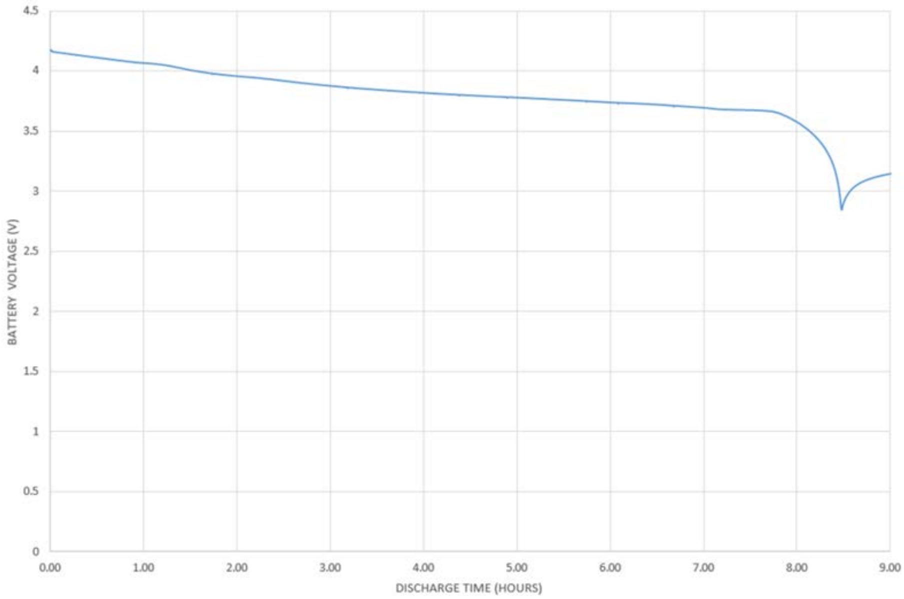
Results and comments

The duration of the main loop is 20 seconds (worst case of the pet tracking application when in "alarm" condition). The battery life in the stated conditions is around 8.5 hours.

Conditions:

- LoRa sends data every 20 seconds
- Data rate is DR_4
- Test is performed indoor (no GNSS fix)

Figure 16. **STEVAL-STRKT01 battery discharge test - full run**



8.2.4 Low power test: 2 minutes, with and without optimizations

Test conditions

We assume that the battery is fully charged (V_{BAT} = 4.2 V). The system goes through the startup phase (startup, LoRa network connection, GNSS fix, sensor reading cycle, transmission followed by low power/ULP mode) before reaching the ultra-low power state.

Four different configuration applicative cases have been taken into consideration and compared as shown in the table below.

Table 4. **Configuration applicative cases**

USB	GNSS cold start	GNSS hot start
Available	Case 3	Case 2
Not available	Case 5	Case 4

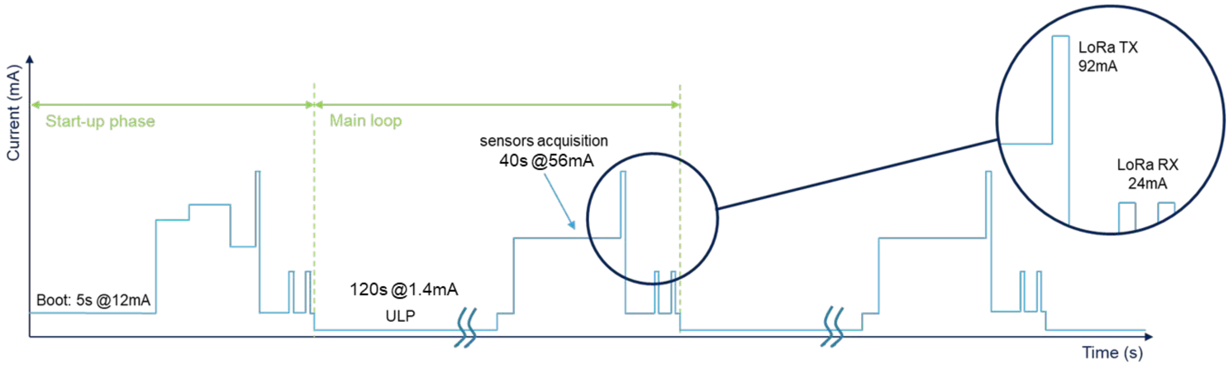
Note: For all configuration applicative cases, see also Figure 12. Low power configuration selection scheme.

The system retrieves data from sensors and GNSS receiver to detect whether the sensor thresholds are overshoot or the GNSS sends geofence messages. The test starts when the Type-C connector is unplugged.

Case 3

Hypotheses related to case 3 are:
- GNSS feeding line is switched on and off
- GNSS executes a cold start at every STEVAL-STRKT01 awakening
- Murata module VDD_TCXO pin is always on
- USB is available for debug

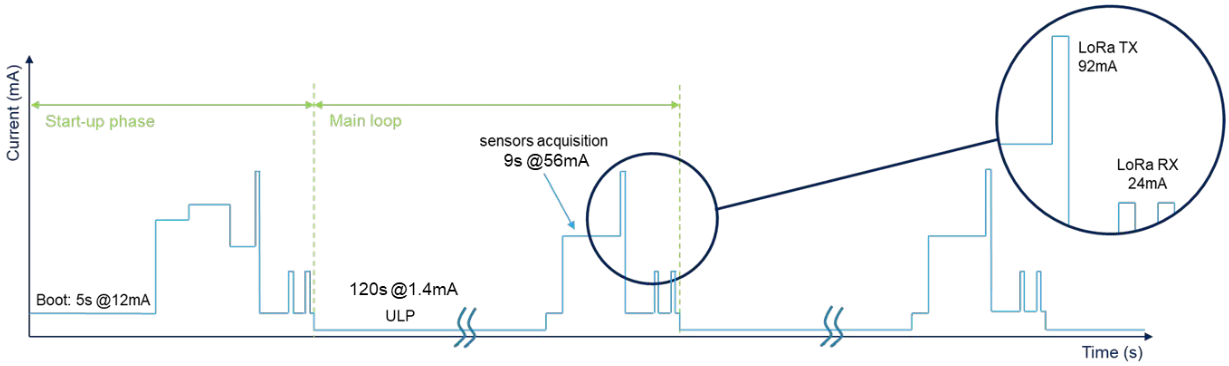
Figure 17. **STEVAL-STRKT01 current consumption profile - case 3**



Case 2

Hypotheses related to case 2 are:
- GNSS feeding line always on
- GNSS executes a hot start at every STEVAL-STRKT01 awakening to speed GNSS fix up (GNSS RTC is on)
- Murata module VDD_TCXO pin is always on
- USB is available for debug

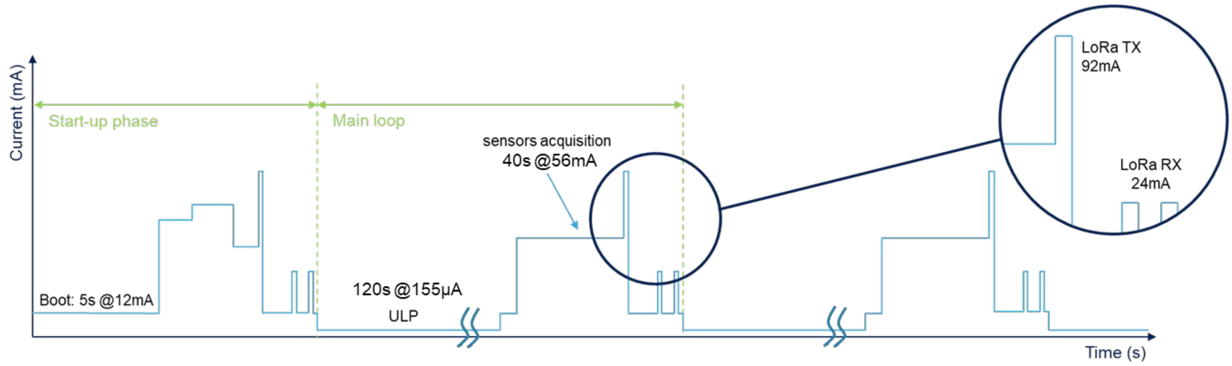
Figure 18. **STEVAL-STRKT01 current consumption profile - case 2**



Case 5

Hypotheses related to case 5 are:
- GNSS feeding line is switched on and off
- GNSS executes a cold start at every STEVAL-STRKT01 awakening
- R107 removed, R106 soldered. TCXO supply line is switched off when the STEVAL-STRKT01 enters ultra-low power mode
- USB is NOT available for debug

Figure 19. **STEVAL-STRKT01 current consumption profile - case 5**

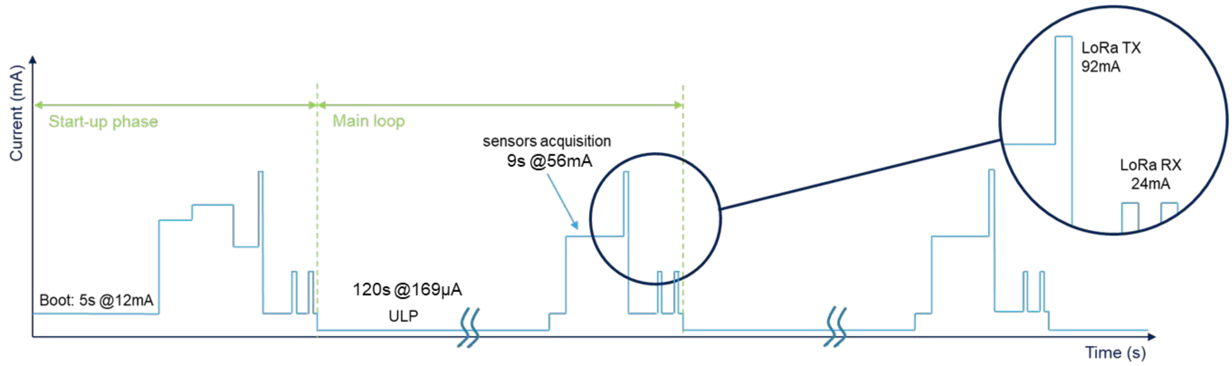


Case 4

Hypotheses related to case 4 are:

- GNSS feeding line always on
- GNSS executes a hot start at every STEVAL-STRKT01 awakening to speed GNSS fix up (GNSS RTC is on)
- R107 removed, R106 soldered. TCXO supply line is switched off when the STEVAL-STRKT01 enters ultra-low power mode
- USB is not available for debug

Figure 20. **STEVAL-STRKT01 current consumption profile - case 4**



Results and comments

The applied hypotheses match well with pet tracking applications. Four tests have been carried out by setting 2 minutes as wakeup frequency. The estimated battery life is around 2 days.

Table 5. **Configuration applicative cases: test results**

USB	GNSS cold start	Results	GNSS hot start	Results
Available	Case 3	Around 38 h	Case 2	Around 43 h
Not available	Case 5	Around 40 h	Case 4	Around 53 h

Figure 21. **Case 3: USB data communication is available (debug mode), awakening every 2 minutes, GNSS cold started**

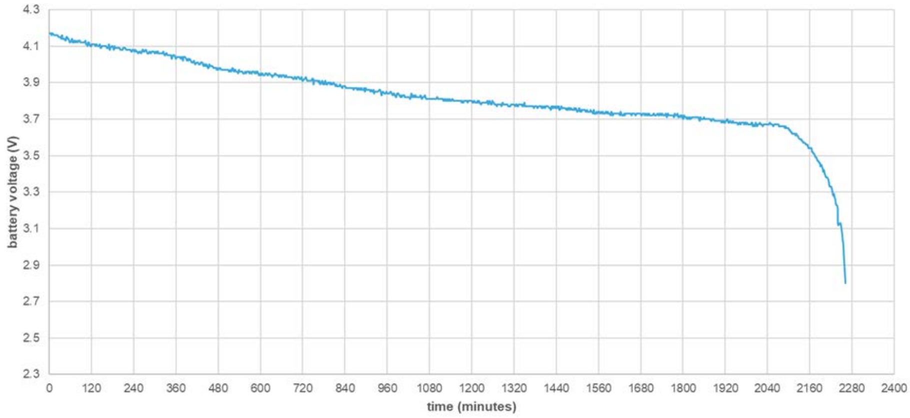


Figure 22. **Case 2: USB data communication is available (debug mode), awakening every 2 minutes, GNSS hot started**

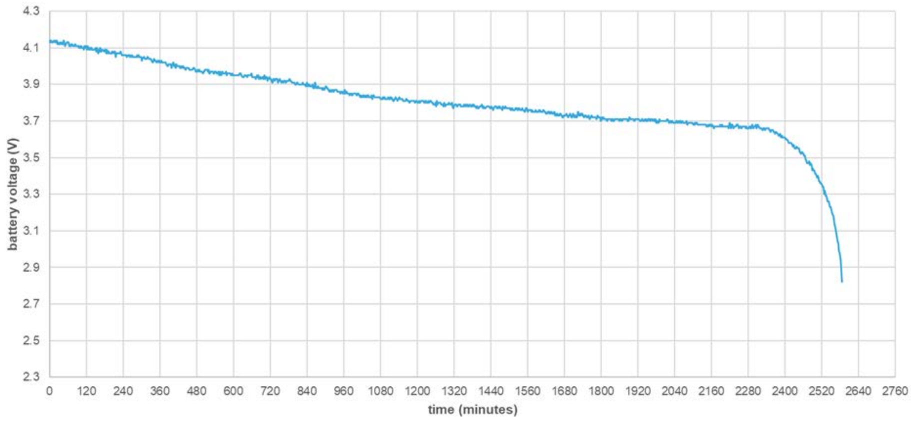


Figure 23. **Case 5: USB data communication is not available (power consumption optimized), awakening every 2 minutes, GNSS cold started**

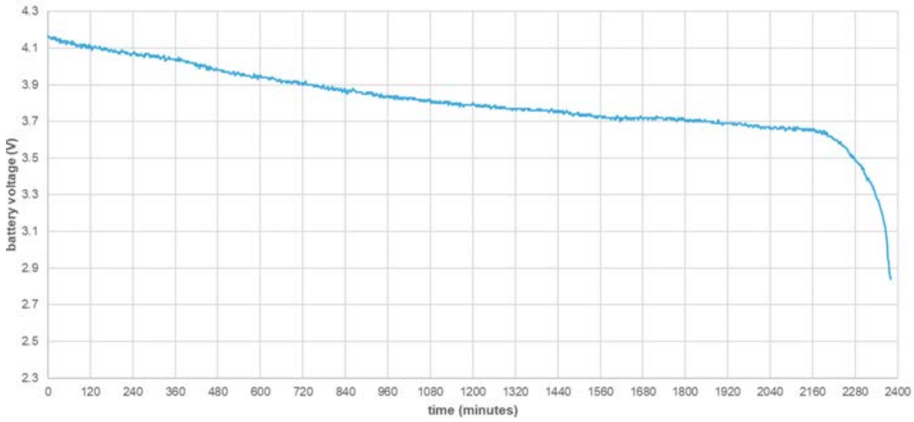
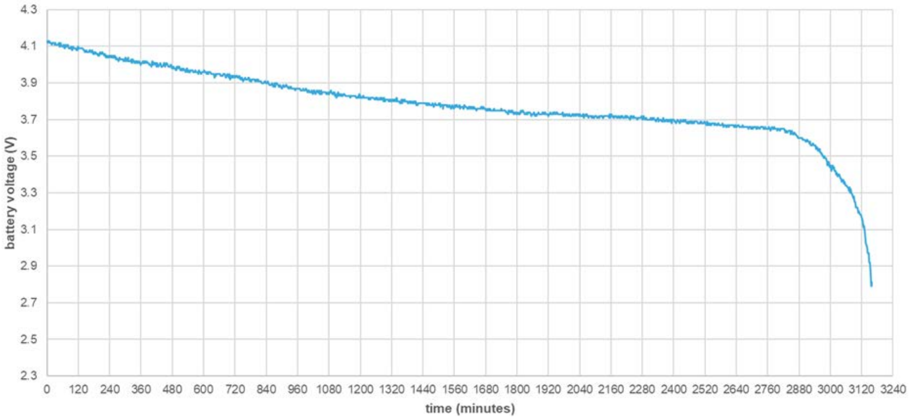


Figure 24. **Case 4: USB data communication is not available (power consumption optimized), awakening every 2 minutes, GNSS hot started**



8.2.5 Low power test: 30 minutes, with hardware optimizations

Test conditions

We assume that the battery is fully charged (V_{BAT} = 4.2 V). The system goes through the startup phase (startup, LoRa network connection, GNSS fix, sensor reading cycle, transmission followed by low power/ULP mode) before reaching the ultra-low power state.

Two different configuration applicative cases have been taken into consideration and compared as shown in the table below.

Table 6. **Configuration applicative cases**

USB	GNSS hot start
Available	Case 2
Not available	Case 4

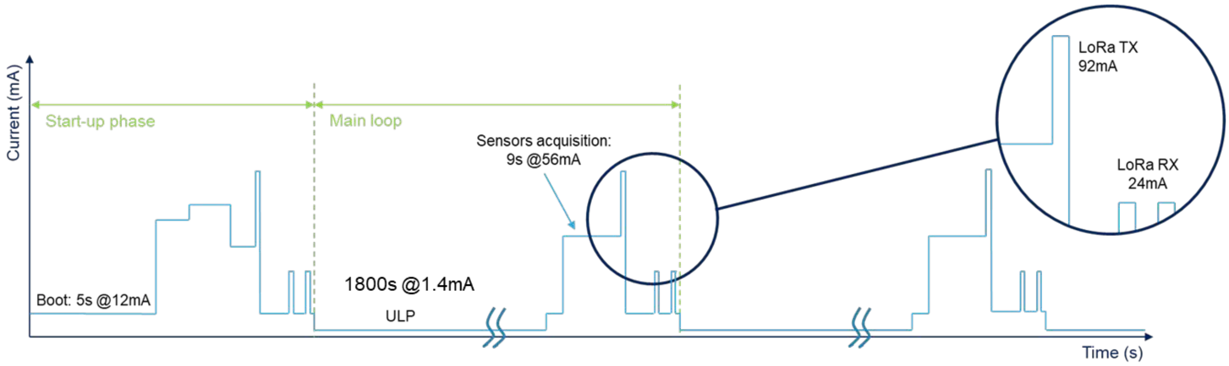
Note: For all configuration applicative cases, see also Figure 12. Low power configuration selection scheme.

The system retrieves data from sensors and GNSS receiver to detect whether the sensor thresholds are overshoot or the GNSS sends geofence messages (see Figure 12. Low power configuration selection scheme).

Case 2

Hypotheses related to case 2 are:
- GNSS feeding line always on
- GNSS executes a hot start at every STEVAL-STRKT01 awakening to speed GNSS fix up (GNSS RTC is on)
- R107 removed, R106 soldered. TCXO supply line is switched off when the STEVAL-STRKT01 enters ultra-low power mode
- USB is available for debug

Figure 25. **Case 2: USB data communication is not available (power consumption optimized), GNSS cold started**

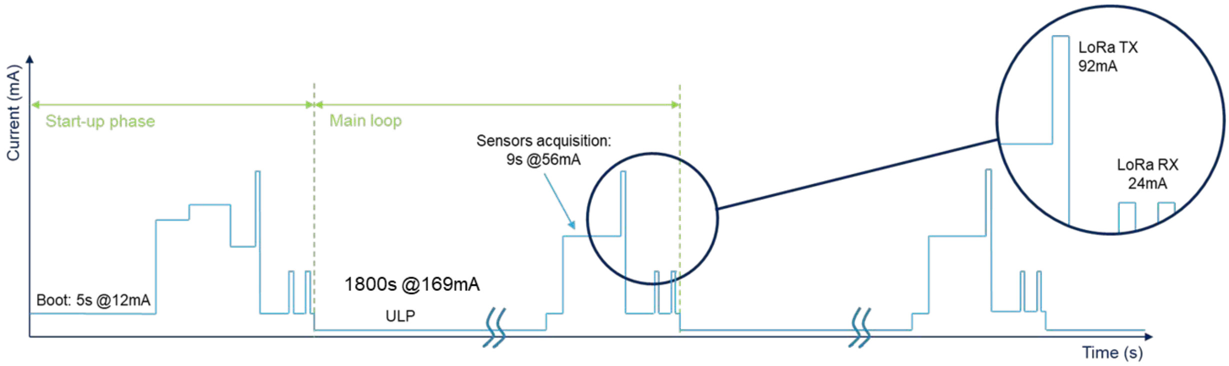


Case 4

Hypotheses related to case 4 are:

- GNSS feeding line always on
- GNSS executes a hot start at every STEVAL-STRKT01 awakening to speed GNSS fix up (GNSS RTC is on)
- R107 removed, R106 soldered. TCXO supply line is switched off when the STEVAL-STRKT01 enters ultra-low power mode
- USB is not available for debug

Figure 26. **Case 4: USB data communication is not available (power consumption optimized), GNSS hot started**



Results and comments

The applied hypotheses matches well with asset tracking applications. Two tests were carried out by setting 30 minutes as wakeup frequency. The estimated battery life is shown in the following table.

Table 7. **Configuration applicative cases: test results**

USB	GNSS hot start	Results
Available	Case 2	Around 213 h
Not available	Case 4	Around 460 h

Figure 27. Case 2: USB data communication is available (debug mode), awakening every 30 minutes, GNSS hot started

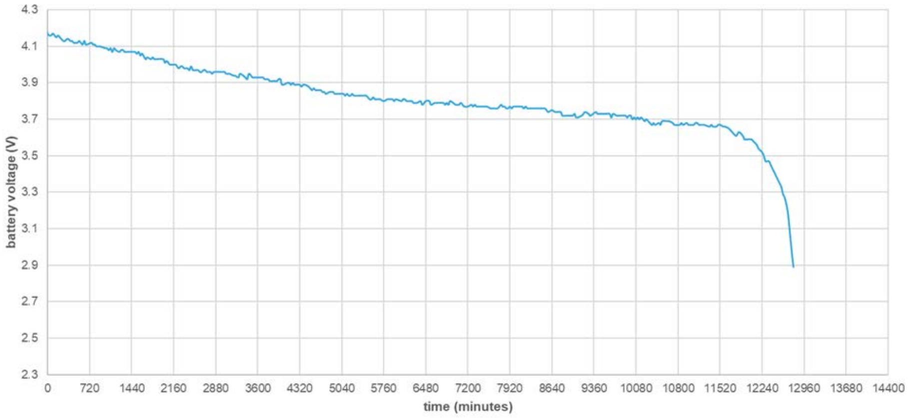
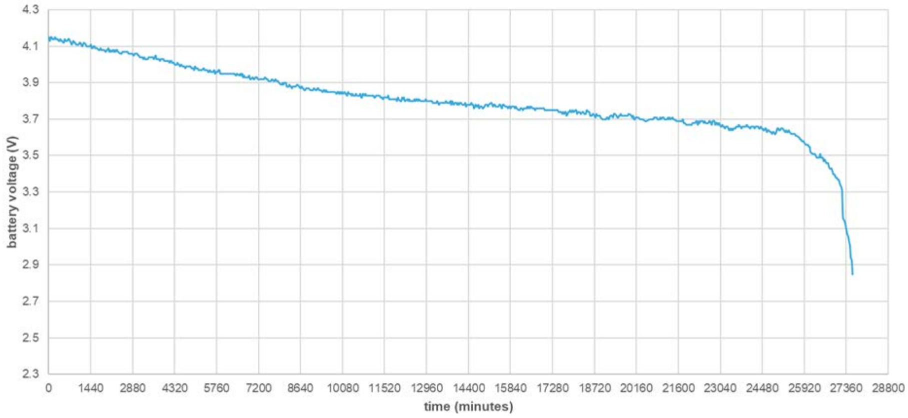


Figure 28. Case 4: USB data communication is not available (power consumption optimized), awakening every 30 minutes, GNSS hot started



9 Schematic diagrams

Figure 29. **STEVAL-STRKT01 circuit schematic (1 of 7)**

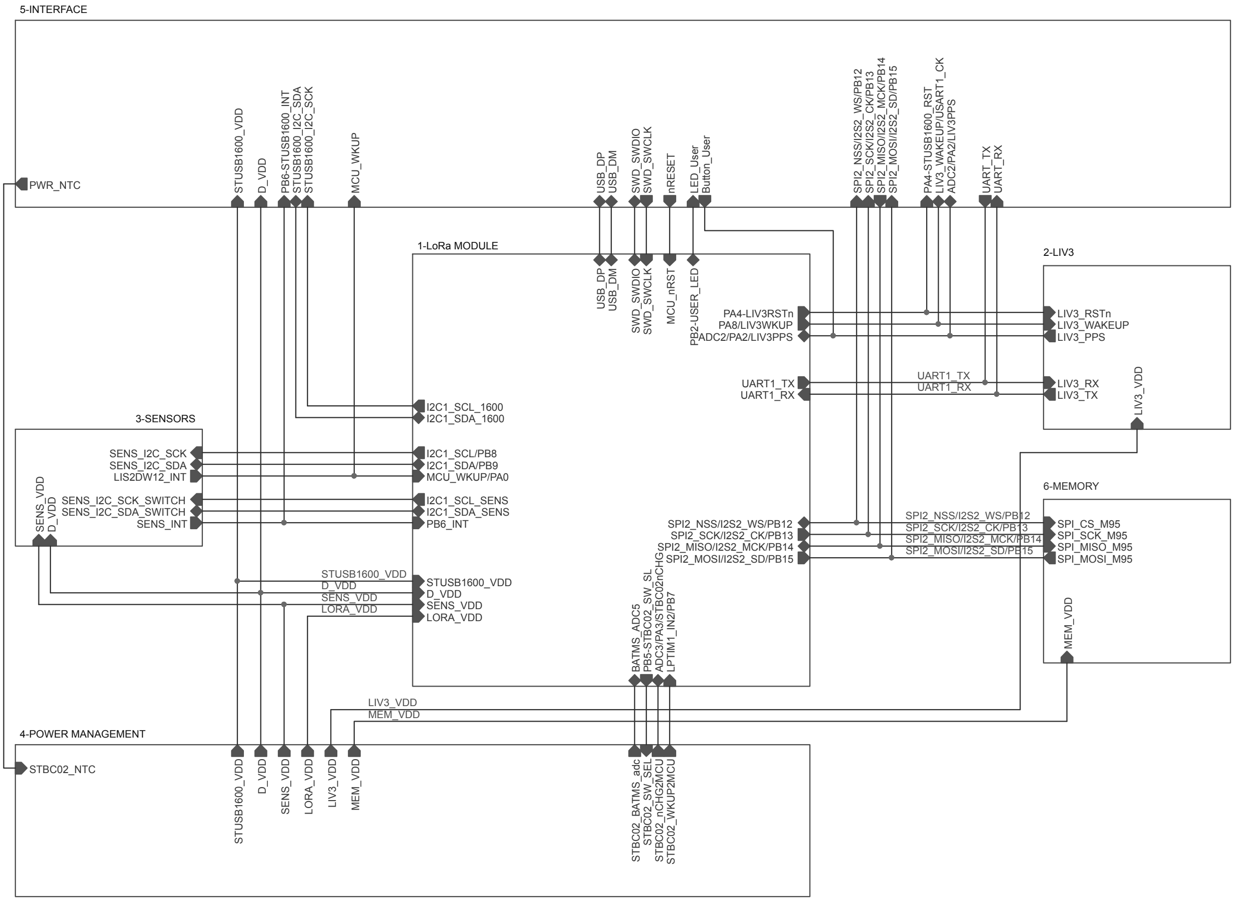


Figure 30. STEVAL-STRKT01 circuit schematic (2 of 7)

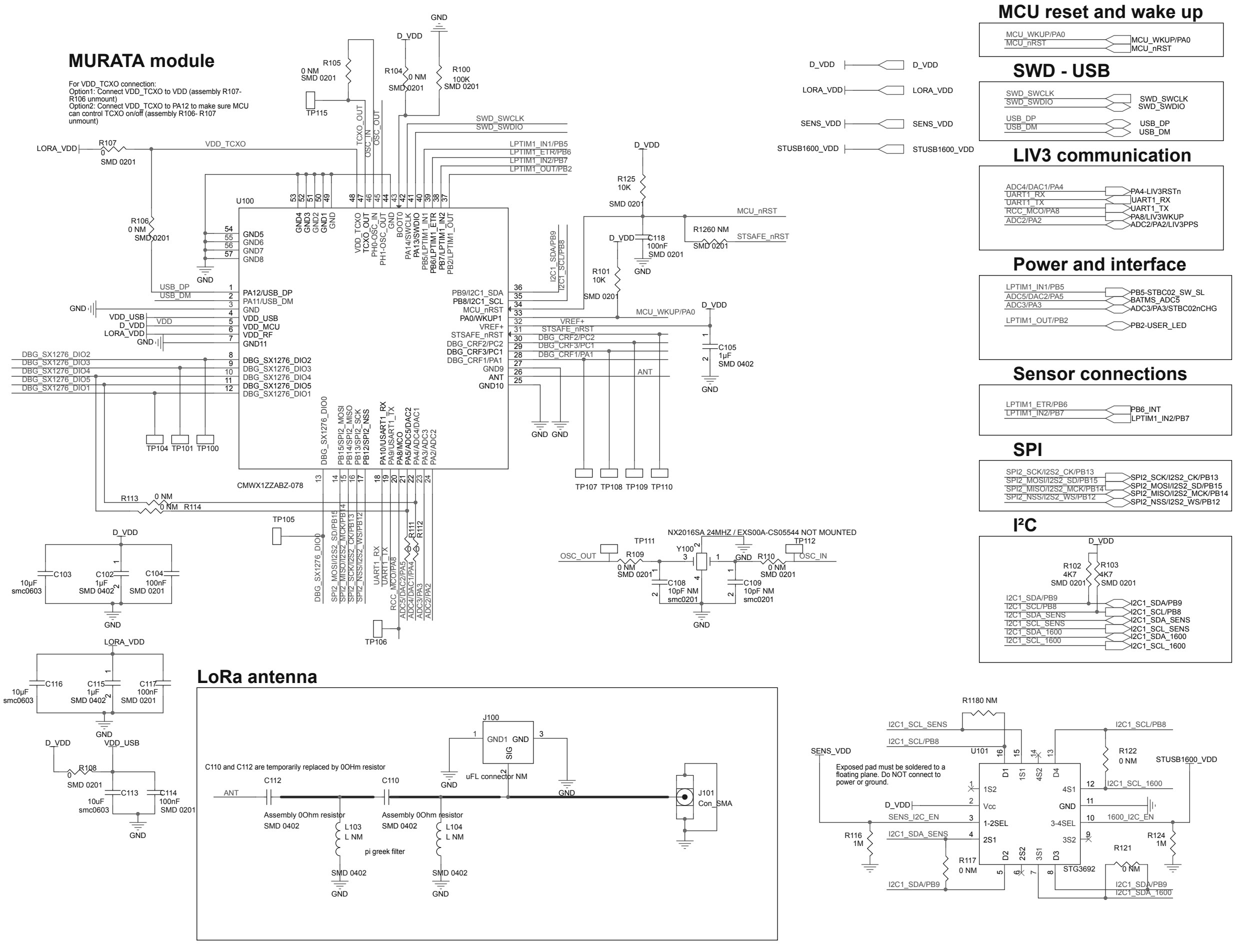


Figure 31. STEVAL-STRKT01 circuit schematic (3 of 7)

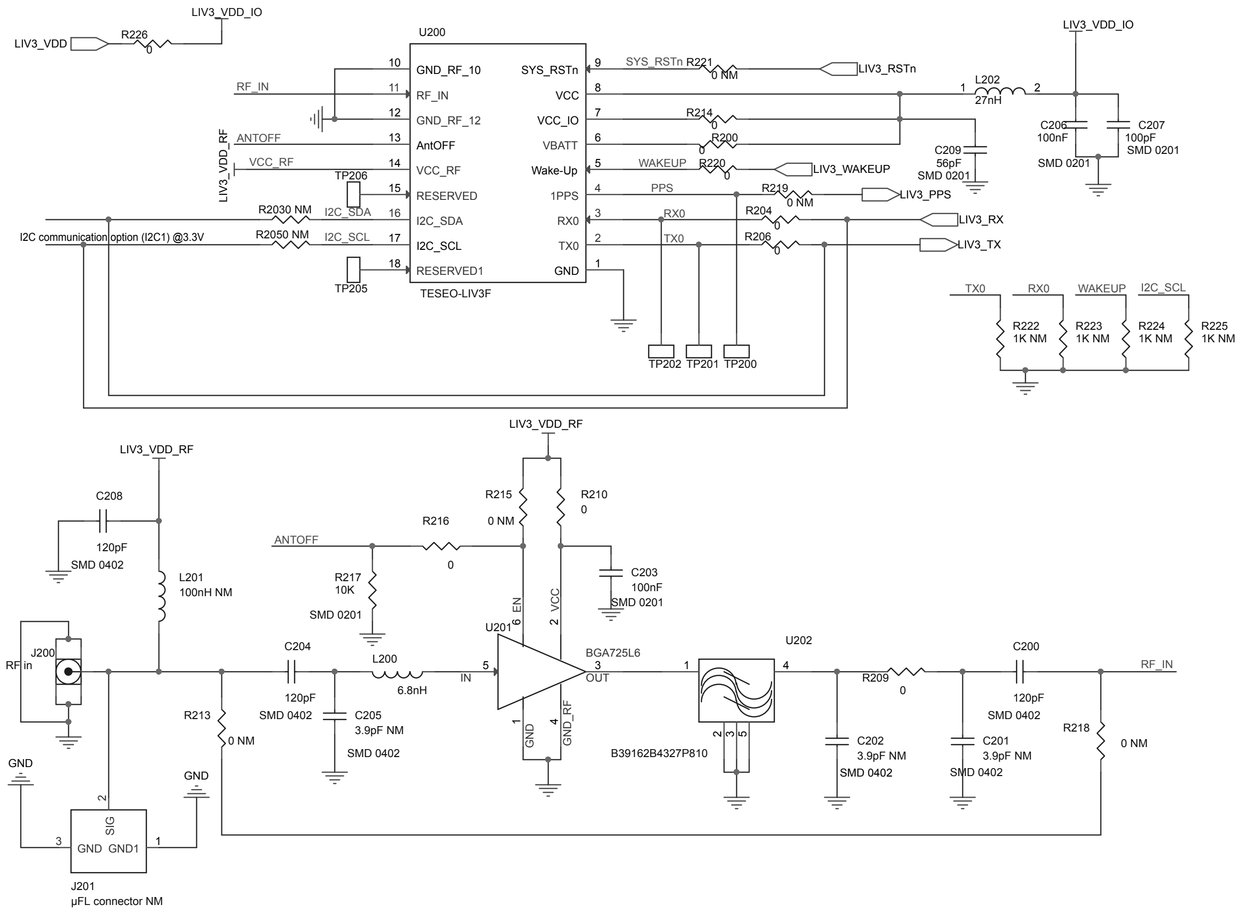


Figure 32. STEVAL-STRKT01 circuit schematic (4 of 7)

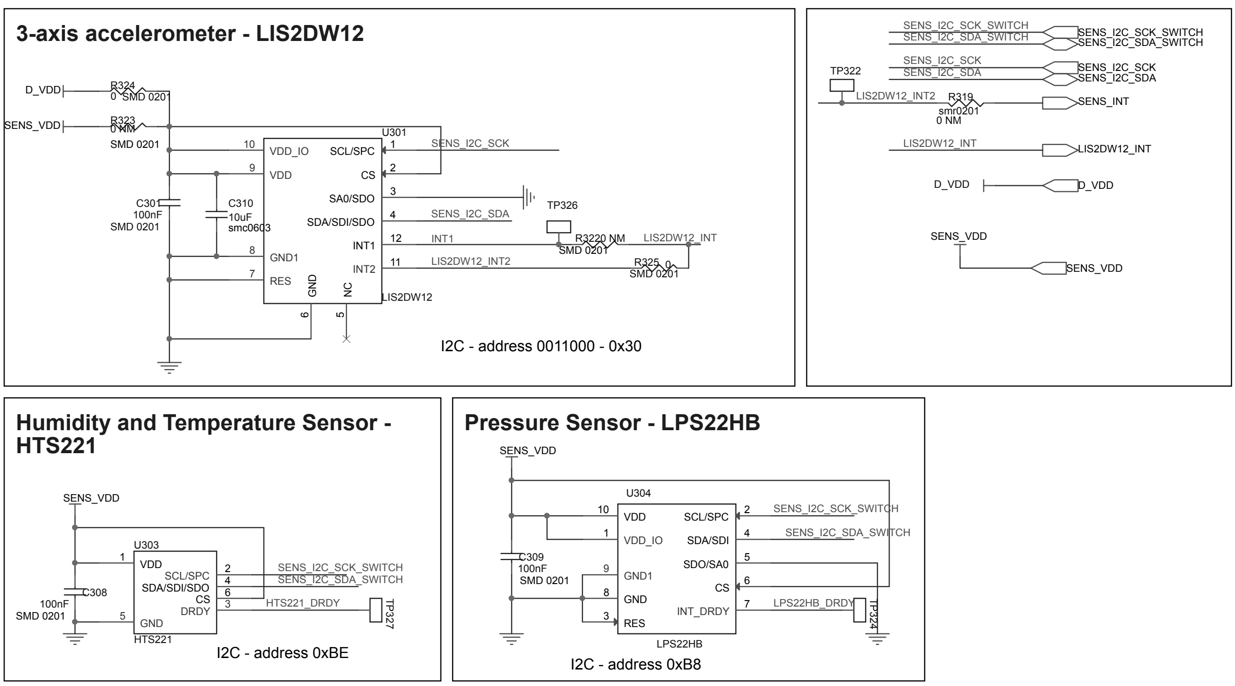
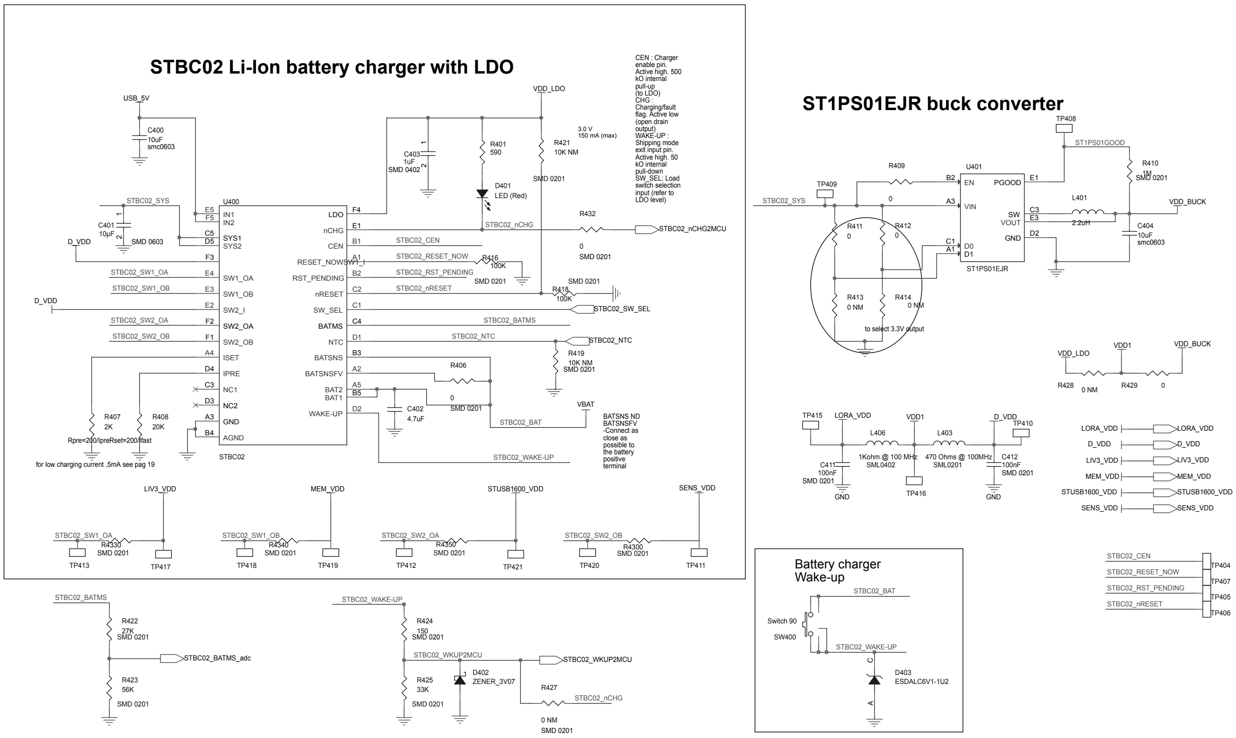


Figure 33. STEVAL-STRKT01 circuit schematic (5 of 7)



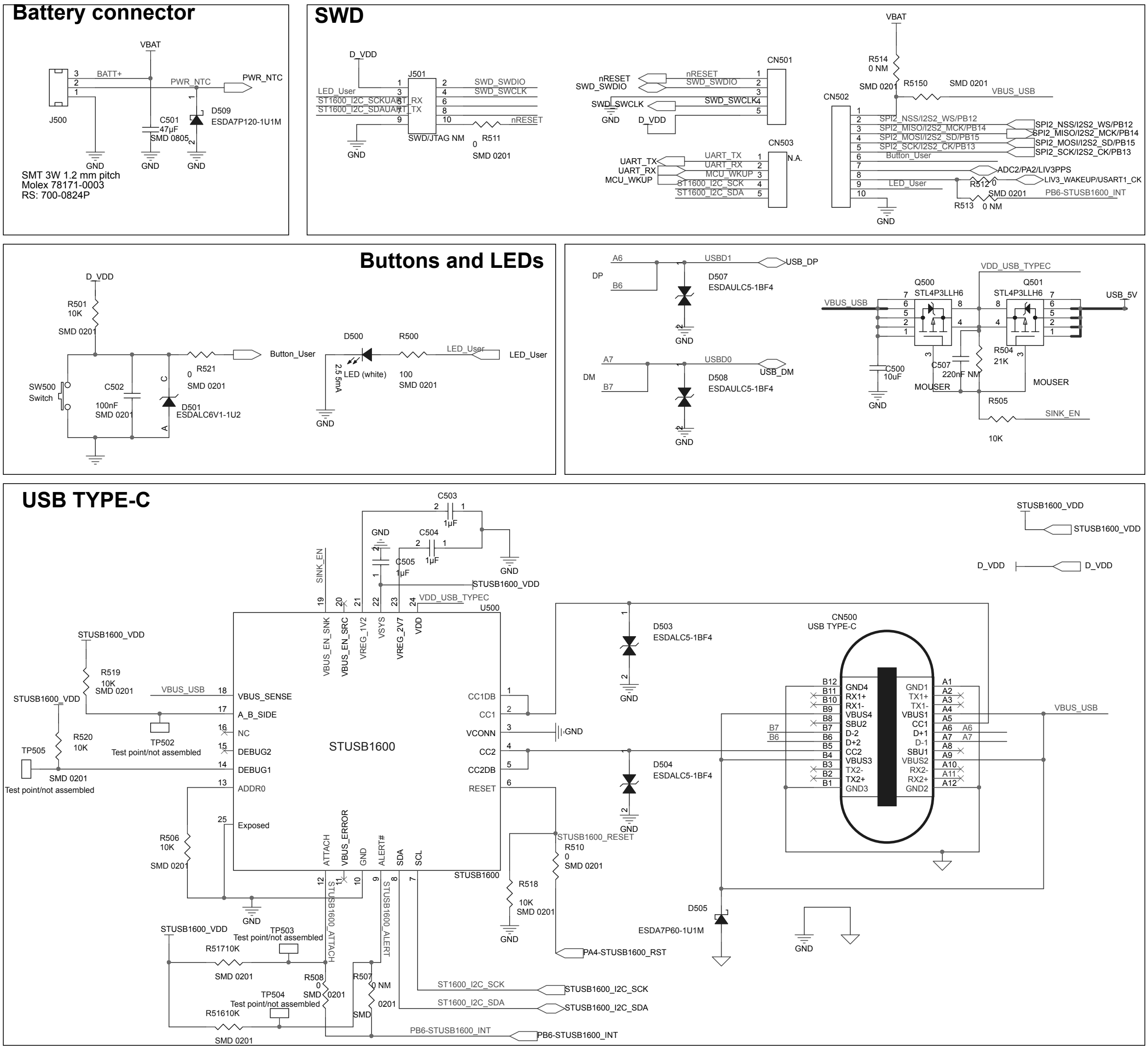


Figure 34. STEVAL-STRKT01 circuit schematic (6 of 7)

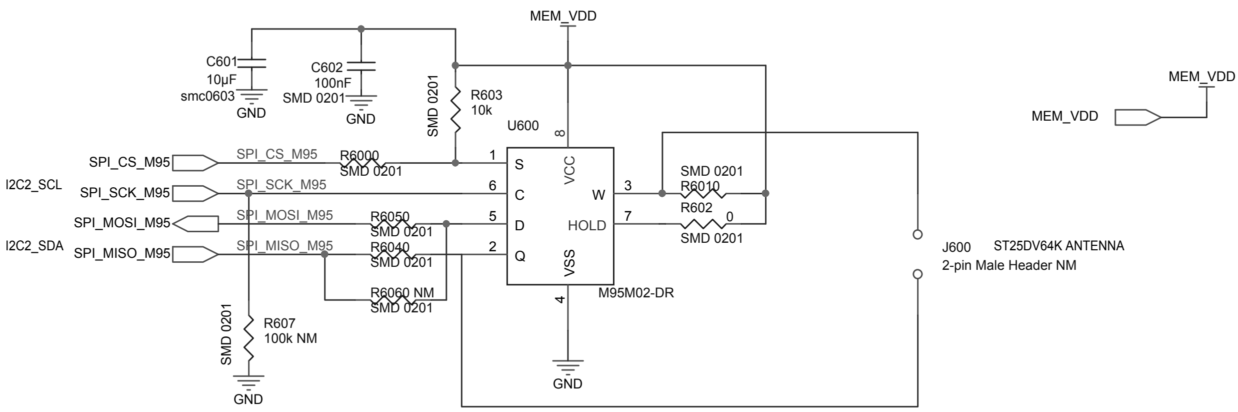


Figure 35. STEVAL-STRKT01 circuit schematic (7 of 7)

10 Conclusions

This application note explains how to select the right configuration to reach the best performance in terms of battery autonomy for the STEVAL-STRKT01.

The most important parameter to be taken into account is the sending frequency: in power consumption optimization, the active components should be operative only when requested by the application ("sleep as much as possible" strategy).

Other configurations can be implemented, such as the low power mode triggered by the inactivity interrupt from accelerometer and ultra-low power mode triggered by a timer.

On the basis of the sending interval, the GNSS power strategy is also impacted to speed its fix process up.

Debug features also impact on the board power consumption. Therefore, they should be activated only if needed.

In summary, depending on the use case, specific configurations can be selected, impacting on the STEVAL-STRKT01 autonomy, resulting in a battery duration of 8.5 hours to 1.4 years.

Revision history

Table 8. **Document revision history**

Date	Version	Changes
28-Oct-2019	1	Initial release.
10-Dec-2019	2	Updated Table 5. Configuration applicative cases: test results.

Contents

List of tables

List of figures