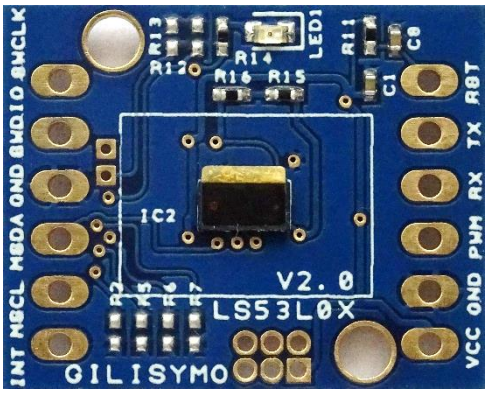
www.gilisymo.com

Gilisymo

LS53L0X-V2 - Low Power Modes Description

Version 2.2.254



Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Gilisymo with respect to the accuracy or use of such information or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Gilisymo's products as critical components in life support systems is not authorized except with express written approval by Gilisymo. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Content

1. Embedded FW Version 3
2. Introduction 3
3. Low Power Mode 3
4. Power consumption in normal mode 5
5. Base mechanism for low power modes 6
6. Low power mode and thresholds 9
6.1. New Threshold functionality 9
6.2. Using new Config parameter Config.UseLowPowerSettings 11
6.3. Using new Config parameter Config.EnableToFLowPower 12
1.1. Using new Config parameter Config.EnableTofWakeup 14
7. Low power modes without threshold 16
8. Standard modes 19

1. Embedded FW Version

This specification is compatible with the following embedded version:

FW version = V2.2.254

You can check the version with the command "**version**" on a UART terminal, see annexes Getting Started Light Saber LS53L0X-V2 board description.

Warning: This version of the Firmware has a typo error on the parameter **EnableTofWakeup** this has been used along all the description because it corresponds to the real parameter. The name will be corrected on next release so be careful on your software about this change.

2. Introduction

This document will describe the low power capability of the LS53L0X-V2. For other functionality please refer to the Light Saber LS53L0X-V2 board description.

The power diagrams contained in that description have been taken with the X-NUCLEO-LPM01A.

Note that all the possible combinations of the parameters have not been tested, it is possible to find some problem. In that case please report it, your feedback is welcome. Please remember that you will always be responsible for your use case.

3. Low Power Mode

As described in the specification, the LS53L0X-V2 supports 10 range modes divided in two categories one for normal mode and one in Low Power mode.

Find hereafter the programming values. Please refer to the specification corresponding to your FW for the supported list. This list corresponds to register from FW version greater or equals to the one reported on previous chapter.

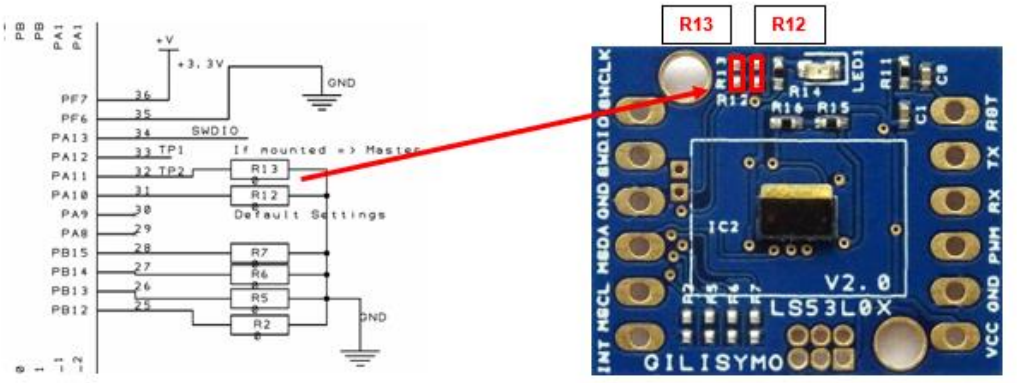
Range Mode	Value	Description
STANDARD_RANGE	**0**	Standard Range as described in the VL53L0X user manual. Timing budget = 30ms. We can range up to 1.2 m.

SHORT_RANGE	**1**	Short Range mode. Timing budget = 30ms. We can range up to 0.5 m.
LONG_RANGE	**2**	Standard Range as described in the VL53L0X user manual. This is the Default ranging mode. Timing budget = 33ms. We can range up to 2 m.
HIGH_SPEED	**3**	High Speed Range as described in the VL53L0X user manual. Timing budget = 20ms. We can range up to 1.2 m.
HIGH_ACCURACY	**4**	High Accuracy Range as described in the VL53L0X user manual. Timing budget = 200ms. We can range up to 1.2 m.
STANDARD_RANGE_LP	**5**	Low Power version of Range Mode 0.
SHORT_RANGE_LP	**6**	Low Power version of Range Mode 1.
LONG_RANGE_LP	**7**	Low Power version of Range Mode 2.
HIGH_SPEED_LP	**8**	Low Power version of Range Mode 3.
HIGH_ACCURACY_LP	**9**	Low Power version of Range Mode 4.

In normal mode the STM32 is always ON. In this mode, user benefits of all the peripherals supported, like: PWM, Master I2C, or Slave I2C, output signals and the UART connection. The latest one permit to drive the Light Saber via serial commands. In normal mode user can drive the board by software.

In low power mode the STM32 will be in standby for a portion of the time. During the time where the STM32 is ON it is possible to drive the Light Saber via software. During the standby portion of the time there is no way to communicate via software with the board. Depending on the period where the STM32 is ON it is possible or not to drive the board via software because there is no time to get command or it is difficult with a terminal to capture this moment. Imagine as example, a low power mode where the LS (Light Saber) do 1 ranging then it waits for 20 hours before wakeup. Do you wait 20 hours to get the chance to stop it? No, for sure. Even after a reset it could be a problem to connect to LS.

The solution in that case is to unlock the LS by hardware. The LS uses a GPIO to detect if user want to uses default settings. In the following image you can see the footprint of the R12 which is a not mounted resistor. With the help of a screwdriver it is possible to short the GPIO to GND, doing so at boot the LS will use the default setting. Note that this is just a temporary configuration because if you reset again the LS it will start with the low power configuration saved. Now you can output from low power configuration, change your parameters if you want, save the parameter and then restart with low power mode.



R12 = configuration bit 0 (config0)
R13 = configuration bit 1 (config1)

Configuration	State
R12 (0 Ohms) mounted	LS53L0X-V2 default settings RECOVERY MODE
R12 (0 Ohms) not mounted (default state)	Use parameters set in flash memory (software configuration mode, USER MODE)
R13 (0 Ohms) mounted	Plugin set as I2C Master
R13 (0 Ohms) not mounted (default state)	Plugin set as I2C Slave

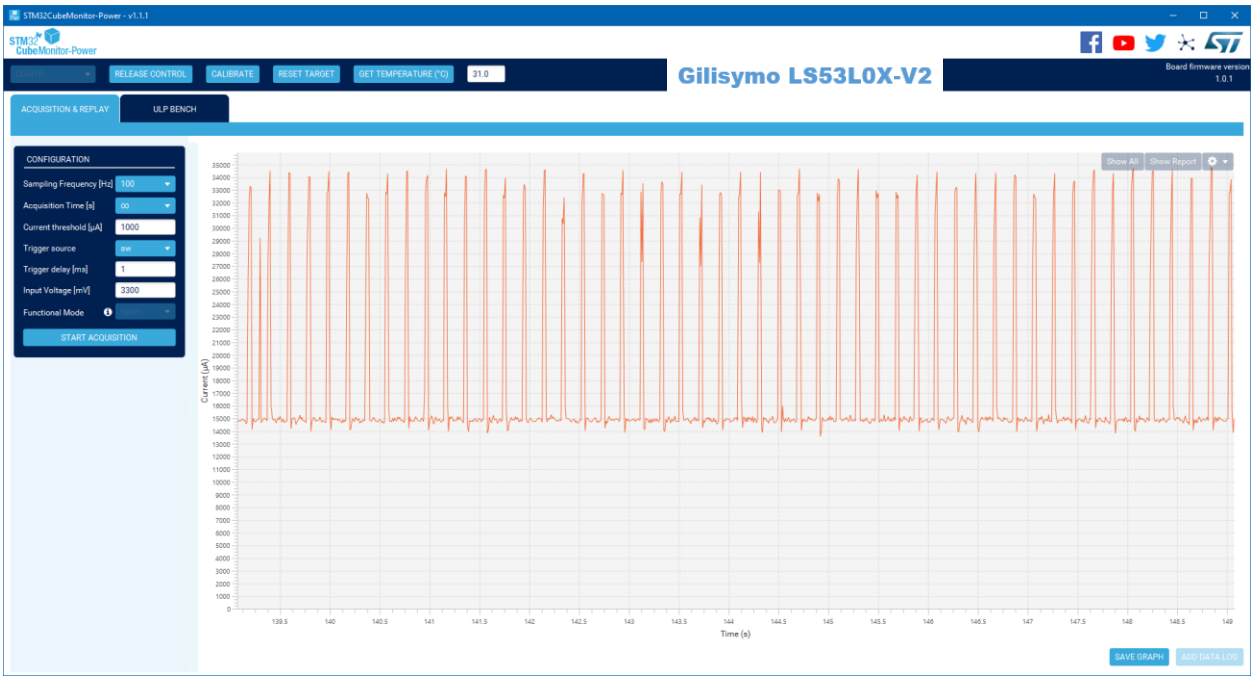
Figure 1: Use a screwdriver to short the R12 then reset the LS to start with Default config

4. Power consumption in normal mode

In normal mode the STM32 is always on. The following power chronogram will show the typical power consumption with a default config:

1) STANDARD RANGE
2) TimingRepeatInMs = 200
3) TimingBudgetInUs = 33000

The STM32 will require 15mA and the peaks represents the consumption of the VL53L0X. The overall consumption will reach 34mA.



5. Base mechanism for low power modes

As described in the LS53L0X-V2 specification, the low power modes follow the scheme in the following picture. Here we can distinguish two phases, one where the STM32 boot, program the VL53L0X then it gets a number of ***StandByCountRanging*** of ranging then for a ***StandByTimeoutSec*** seconds both the STM32 and VL53L0X will remains in standby. This sequence will be repeated indefinitely.

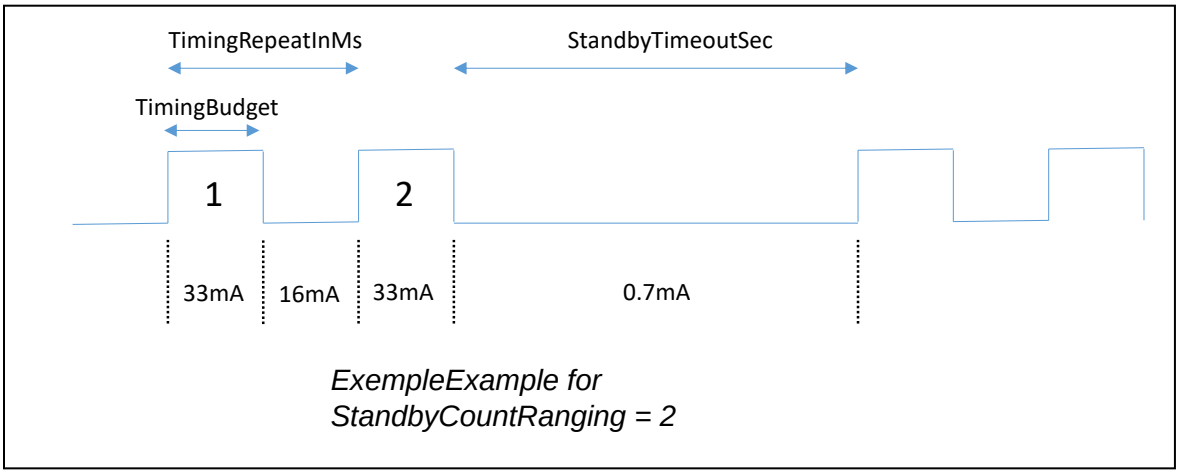


Figure 2: Low Power example

The following power chronogram will show the typical power consumption with the following config:

1) STANDARD_RANGE_LP
2) TimingRepeatInMs = 19
3) TimingBudgetInUs = 19000
4) StandByCountRanging = 5
5) StandByTimeoutSec = 5

Note that when you modify the range mode all the other parameters are set as the default. This means that you need to **set prior the range mode** and then all the other params.

I reduced both StandByCountRanging and StandByTimeoutSec for better visibility on the chronogram.

- set Config.RangeMode=5
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- savepar

The following chronogram has been taken from the previous FW 2.0.222. The STM32 will require around 15mA and the peaks represents the consumption of the VL53L0X. The overall consumption will reach 34mA. For around 5 seconds the power consumption is very low under 0.7mA

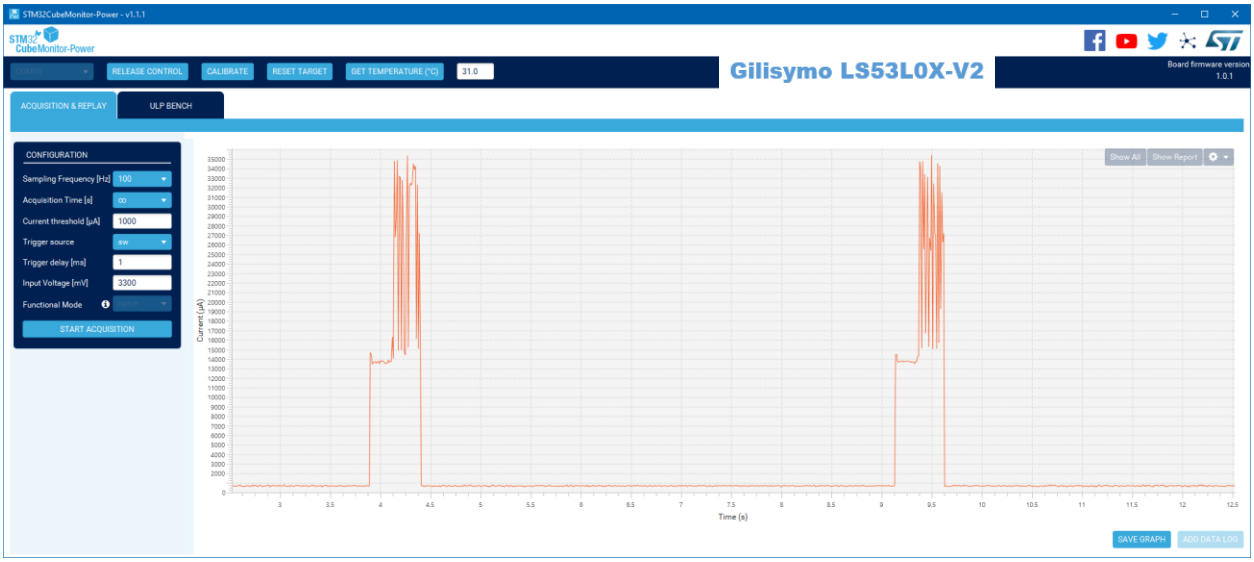


Figure 3: Power Chronogram with version v2.0.222

The new FW will reduce more the power consumption in standby. The FW 2.0.222 in standby has a 0.7 mA of consumption whilst the new one 2.2.254 is under 20 uA.

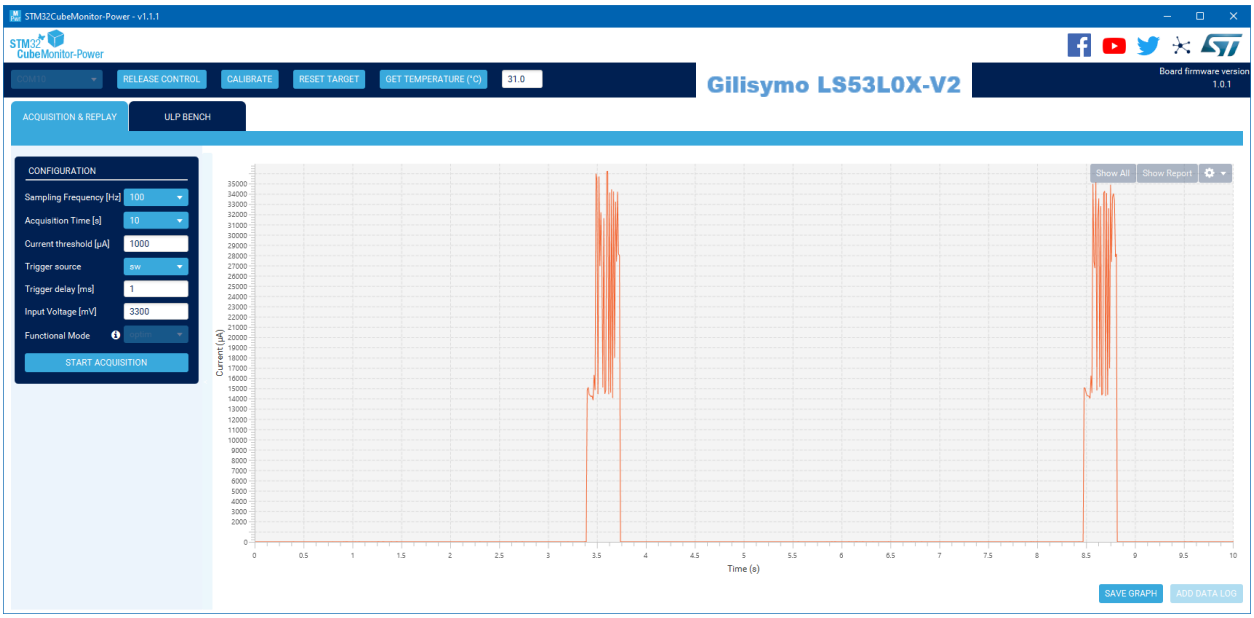


Figure 4: Power Chronogram with version v2.2.254

The following power chronogram is a detail around the first wakeup.

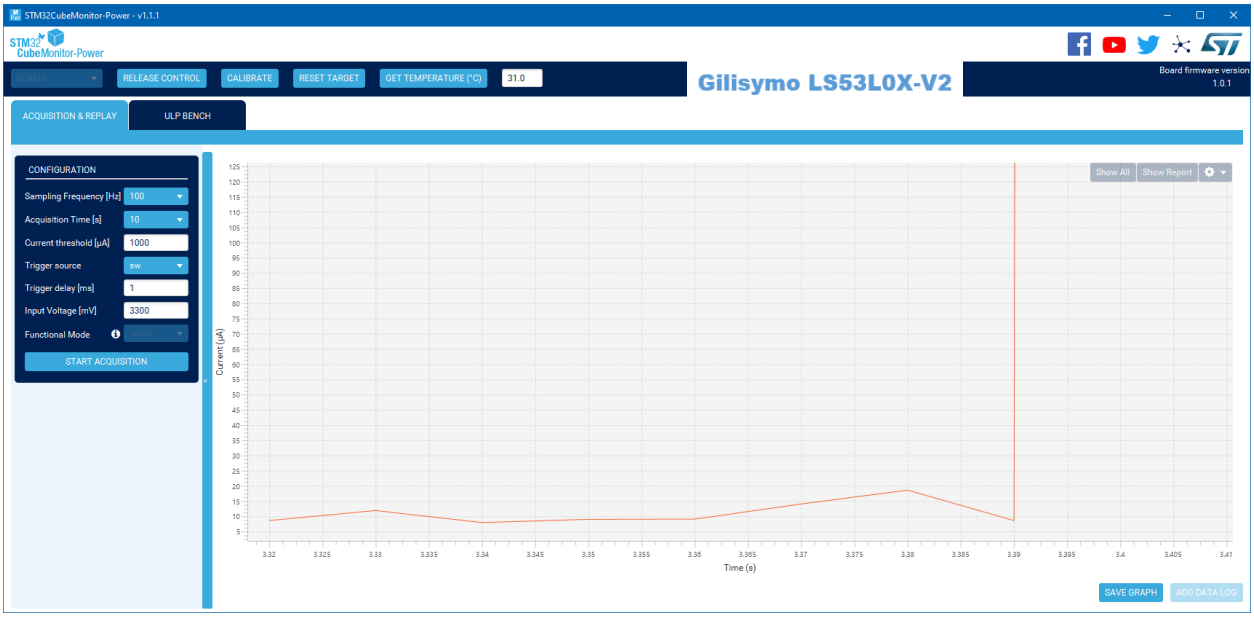


Figure 5: Power Chronogram with version v2.2.254 details

6. Low power mode and thresholds

6.1. New Threshold functionality

The Light Saber supports the threshold mechanism, for more details please refer to the Light Saber LS53L0X-V2 board description.

When the threshold is selected, the LS will provide the output only if the ranging satisfies the thresholds conditions. For example, the ranging distance is below a given value, is greater than a value or is in a range.

In normal ranging modes the STM32 is always ON the advantages of the thresholds is for the host connected to the interrupt output which can reduce its power. However, for the LS there is no benefit on the power consumption.

In case of low power modes, range modes with _LP, we already benefit of the power reduction in the standby phase. With these low power modes, however, the threshold is tested only when the STM32 is ON.

The new FW improves the power consumptions in thresholds modes.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- **set MinThreshold=100**
- savepar

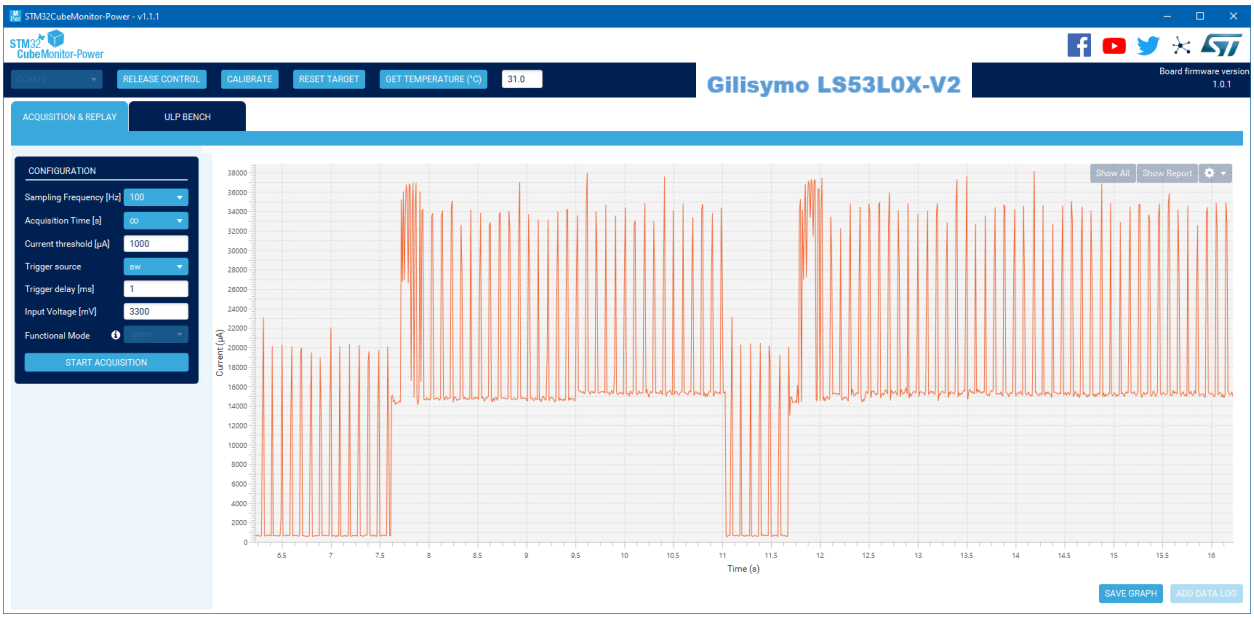


Figure 6: Power Chronogram with MinThreshold=100

When the threshold is enabled the new FW will follow the base scheme as in the chapter 5. This time the VL53L0X continue the ranging, you can see the peaks of power with low power. When an object is present under the threshold in this case the 100mm the STM32 wakeup and continue to ranging but with high power. It stays for a while then if no more objects under the threshold the STM32 go to standby again.

The advantage here is that if no object detected in the period of StandByTimeoutSec at least the consumption of the STM32 is saved. The more is the value of StandByTimeoutSec the more is the power saving.

In that configuration the minimum power consumption is around 0.7mA.

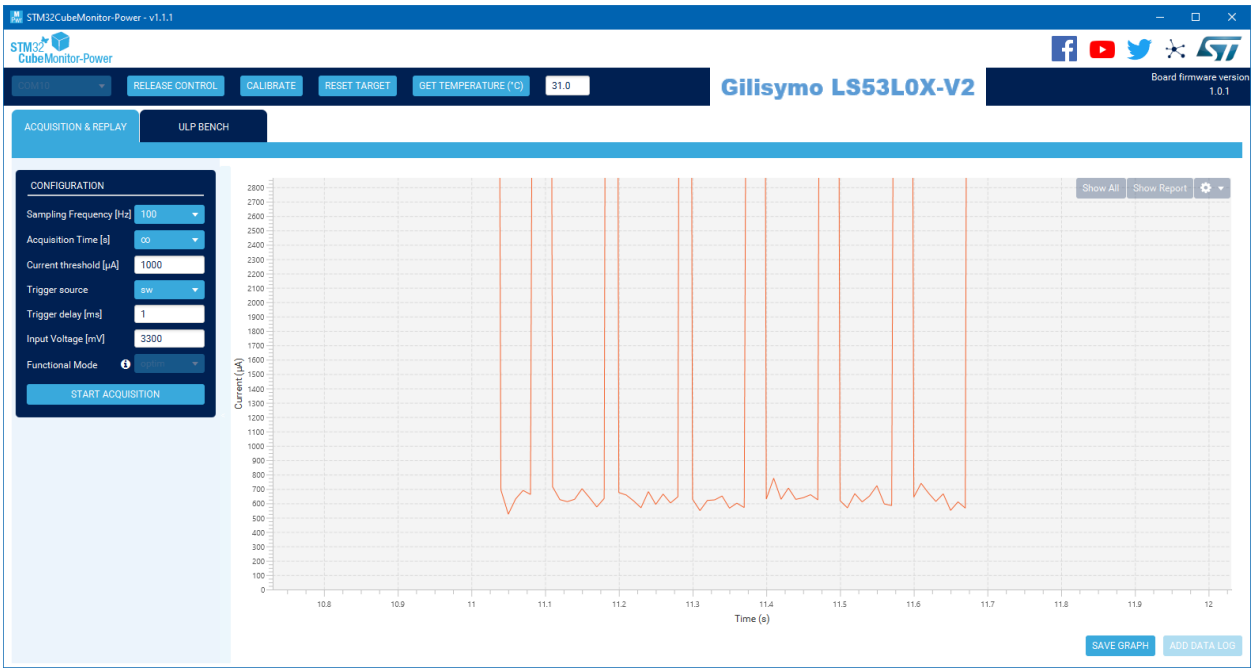


Figure 7: Power Chronogram with MinThreshold=100 detail

6.2. Using new Config parameter Config.UseLowPowerSettings

When you enable this setting the power consumption from the VL53L0X will be reduced, the result of ranging is less precise but if you need only to detect an object this can be a good choice and low power.

In the following power chronogram, you can see that in standby the Monitor-Power has some difficulties to get the signal from power consumption. For sure we can increase the resolution on the GUI but here we can observe the difference compared with the normal case.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- set MinThreshold=100
- **set Config.UseLowPowerSettings=1**
- savepar

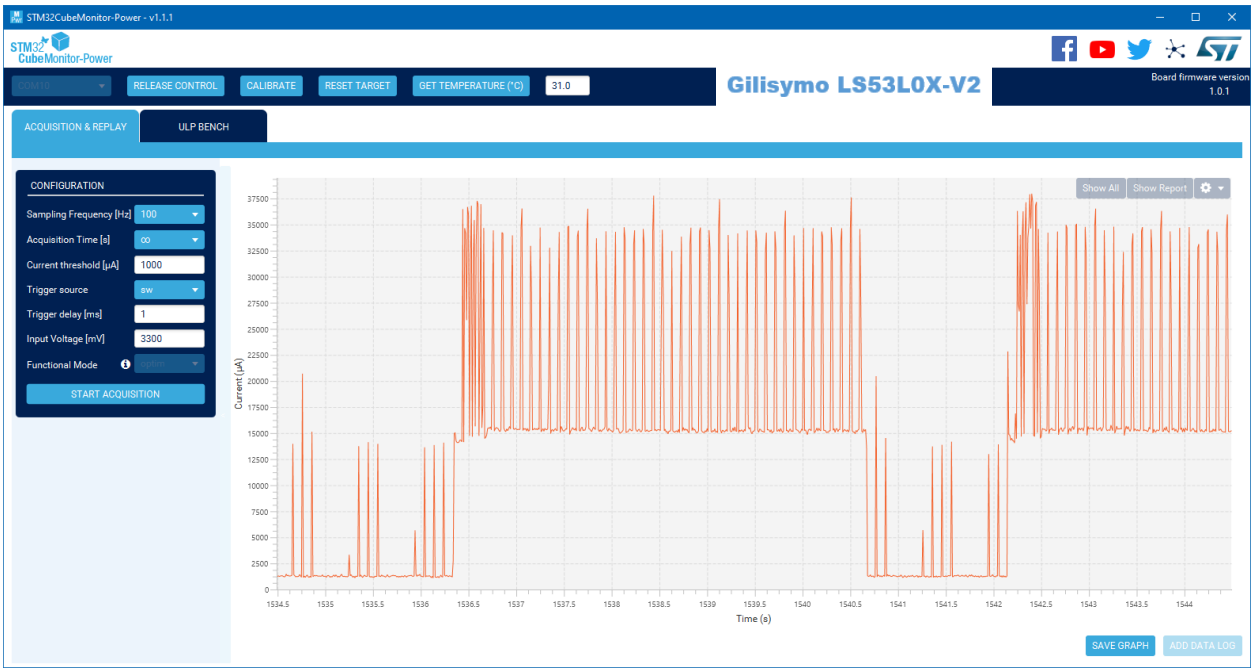


Figure 8: Power Chronogram with MinThreshold=100 with UseLowPowerSettings=1

6.3. Using new Config parameter Config.EnableToFLowPower

When you enable this setting the power consumption from the VL53L0X will be reduced more. The ranging is less precise due to Config.UseLowPowerSettings=1 but this new setting will put in standby the STM32 immediately after it gets data from VL53L0X and send data over the UART, except the first time when the STM32 program and launch the ranging. The timing used for the VL53L0X is a back to back so ranging on VL53L0X works without the help of the STM32.

In the following power chronogram, you can see that the STM32 is most of the time in standby, the power of VL53L0X is reduced and a threshold is used. This means that as soon as an object is detected under the threshold an interrupt is sent to the STM32 that wakeup but it just read values from VL53L0X send data over UART then go into standby again.

At the end of the chronogram you can see that the STM32 wakeup, this is due to the fact that at the same time of the interrupt from VL53L0X you can wakeup with a timer and the timer is set with the StandByTimeoutSec. This is useful to define a kind of watchdogs, in fact when it wakeup from standby from the timer, the STM32 will reprogram the VL53L0X and after a number of StandByCountRanging it goes again in standby.

The period for the ranging is given by the TimingRepeatInMs so if this is 100ms and the TimingBudgetInUs=20000us you obtain a 20ms ranging and 80ms standby with a power spent by the STM32 to read and send data over UART.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- set MinThreshold=100
- **set Config.UseLowPowerSettings=1**
- **set Config.EnableToFLowPower=1**
- savepar

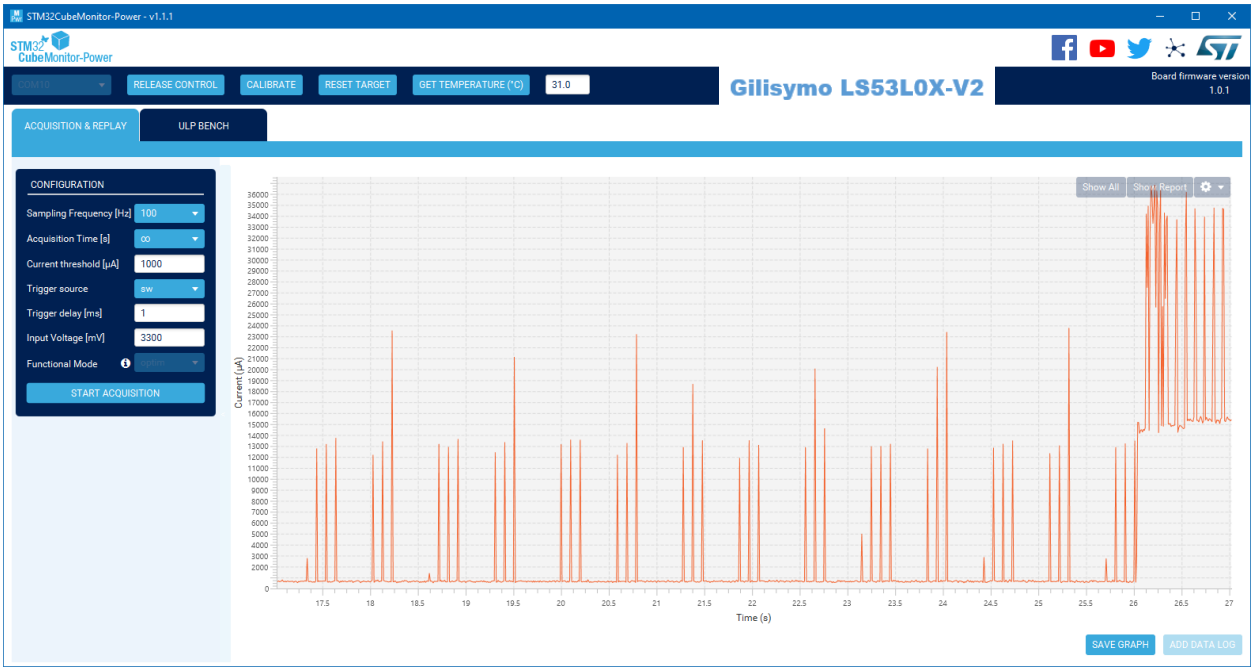
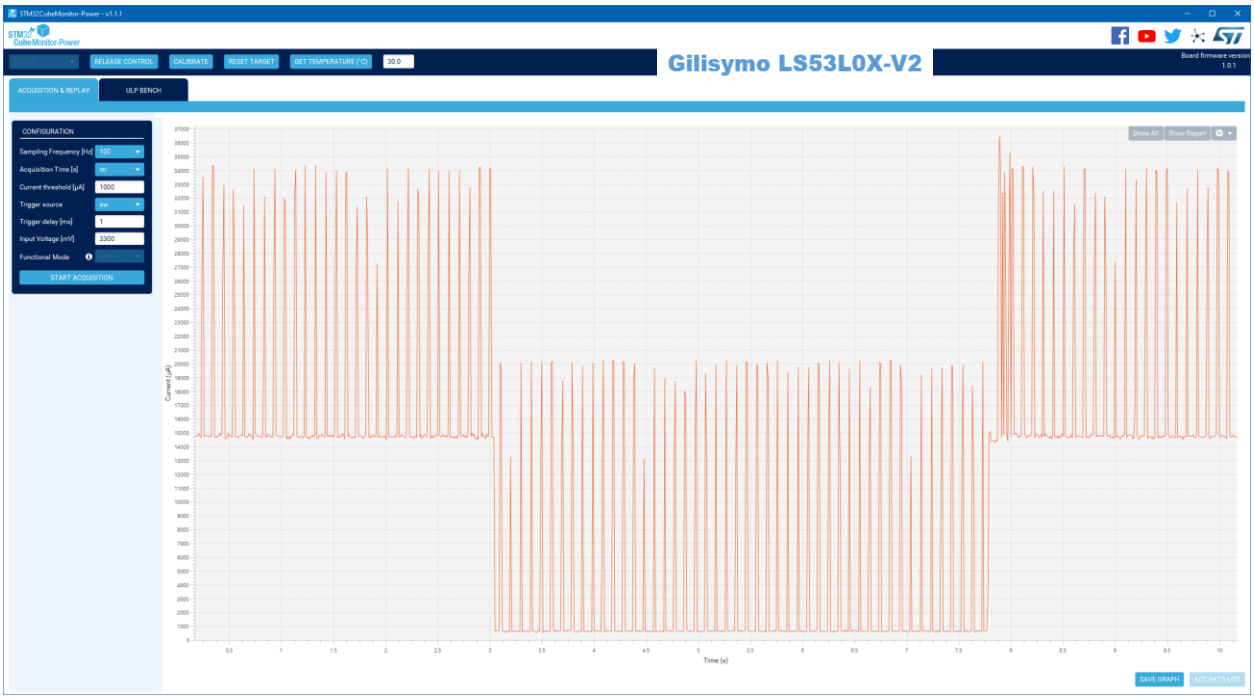


Figure 9: Power Chronogram with EnableToFLowPower=1

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- set MinThreshold=100
- **set Config.EnableToFLowPower=1**
- savepar



1.1. Using new Config parameter Config.EnableTofWakeup

When you enable this setting, the STM32 can wakeup only with the VL53L0X interrupt in this case the RTC interrupt is disabled. This permit to remain in low power condition and in case of threshold only when an object is detected the STM32 wakeup. The counterpart is that we cannot drive the LS by software but only by hardware. In that case if it is possible to satisfy the threshold condition it is easy to wake up and drive again by software.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- set MinThreshold=100
- **set Config.UseLowPowerSettings=1**
- **set Config.EnableTofWakeup=1**
- savepar

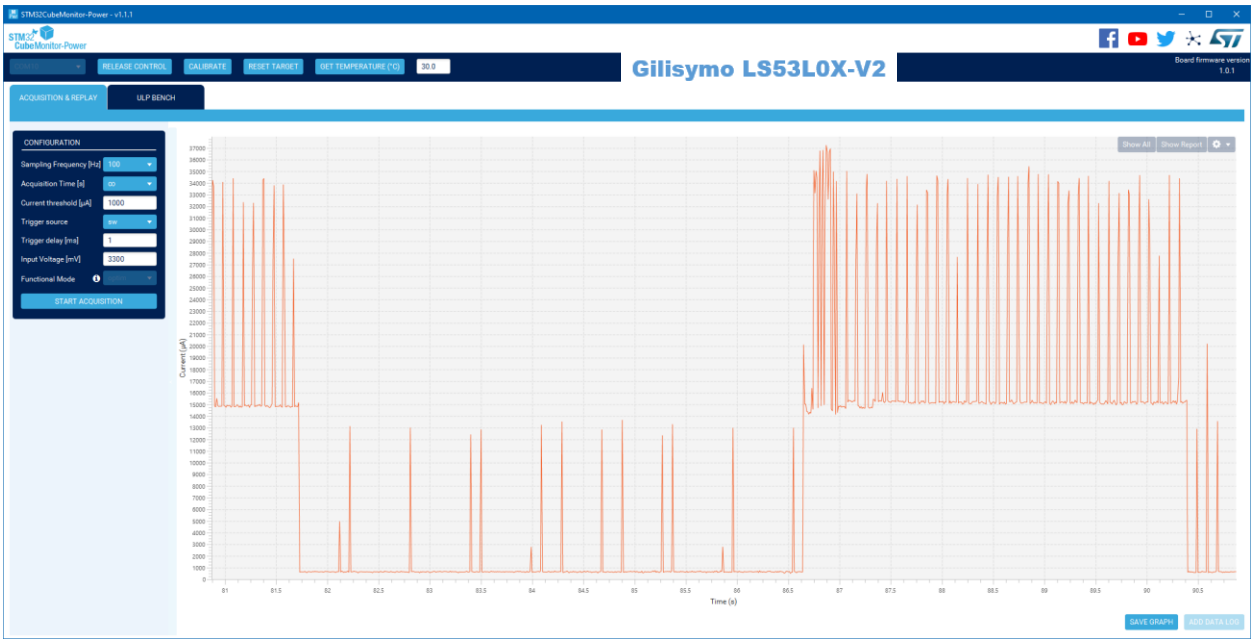
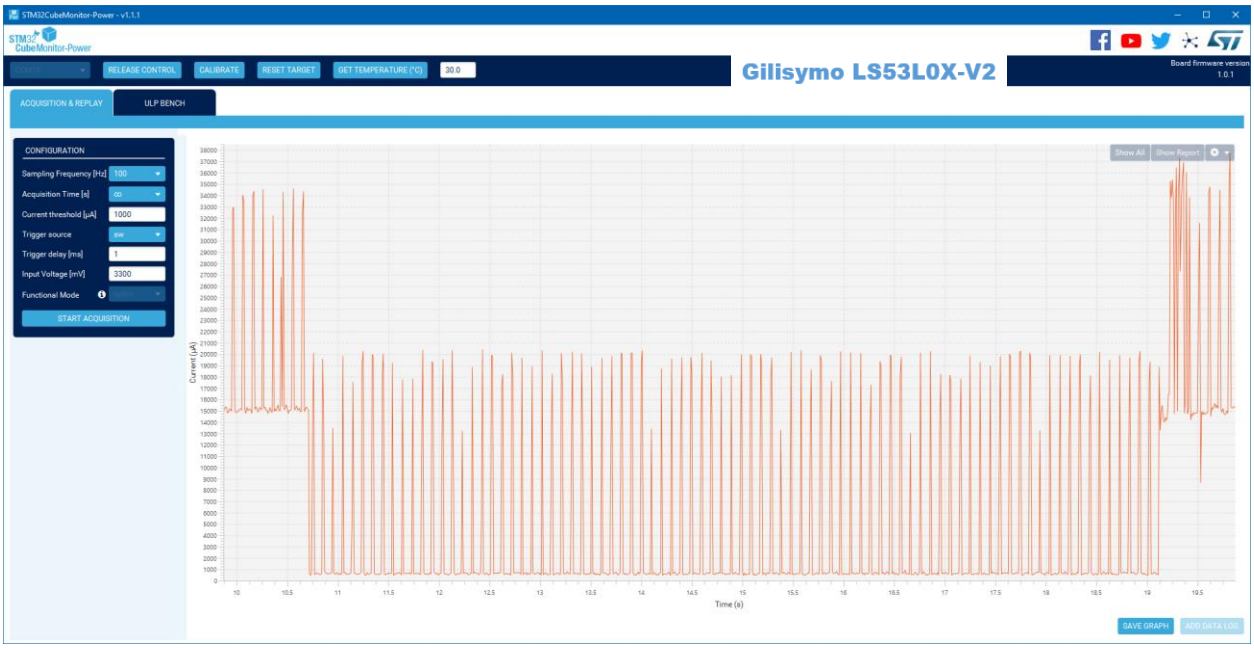


Figure 10: Power Chronogram with EnableTofWakeup=1

The chronogram seems the same as before but the standby parts stay for more times and only the presence of my hand make it wake up and go to normal mode for StandByCountRanging ranging then it goes again to standby.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- set MinThreshold=100
- **set Config.EnableTofWakeup=1**
- savepar



In that case only an object will wake up the STM32 but the VL53L0X uses its max power. This is useful in case of high ambient light or without a threshold to go far.

7. Low power modes without threshold

The following power chronogram corresponds to a configuration in low power modes i.e. from 5 <= RangeMode <= 10, without thresholds.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- **set Config.EnableTofWakeup=1**
- savepar

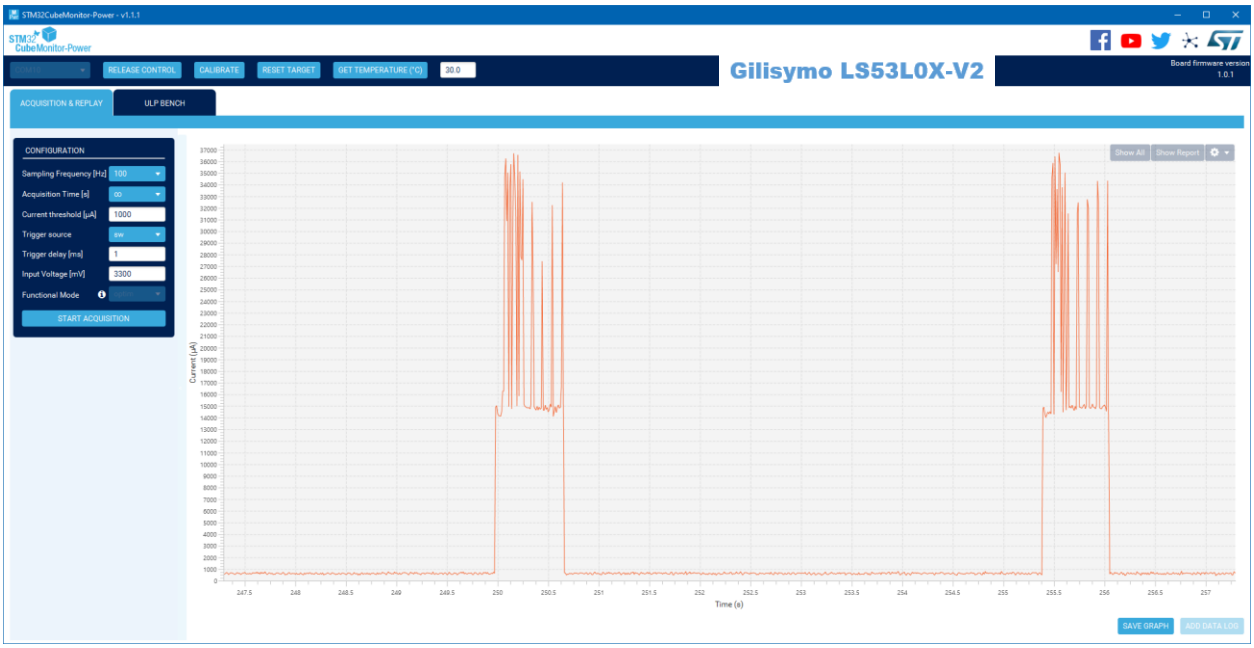


Figure 11: Power Chronogram with EnableTofWakeup=1 no threshold

In that case only the new flag doesn't have any effect because since no threshold activated the ranging is stopped by the STM32 before going to standby.

With the Config.EnableToFLowPower=1, when the LS go to standby, it will not turn off the ranging on VL53L0X.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- **set Config.EnableTofWakeup=1**
- **set Config.EnableToFLowPower=1**
- savepar

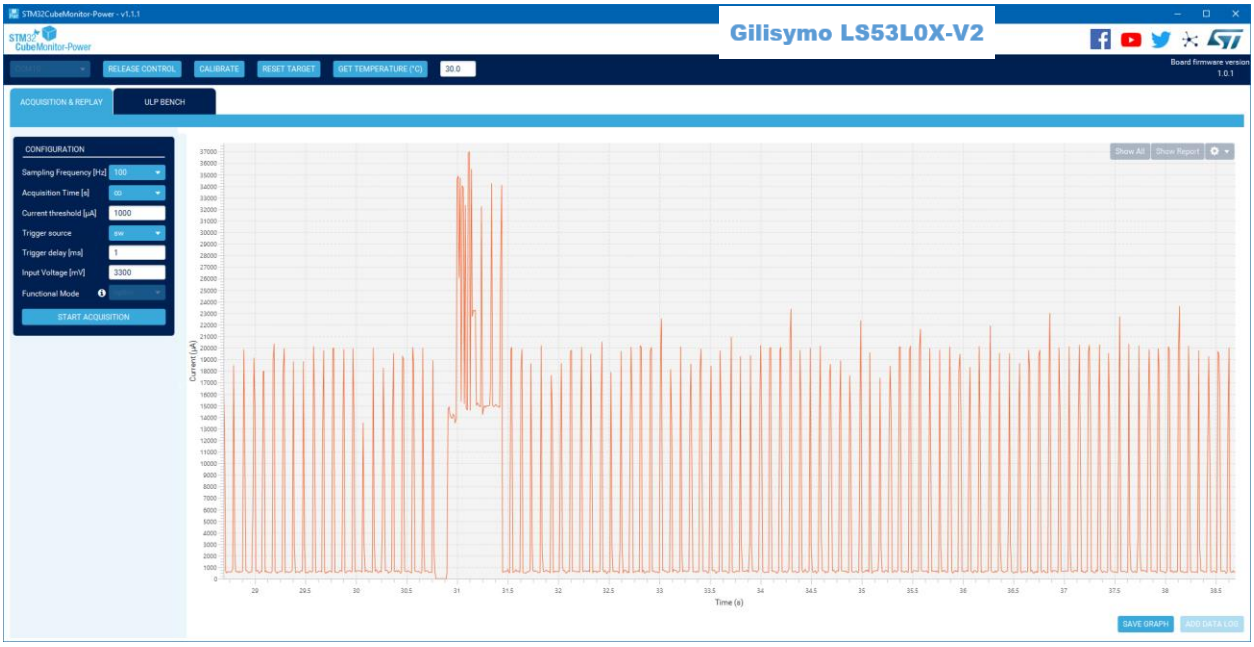


Figure 12: Power Chronogram with EnableTofWakeup=1 EnableToFLowPower=1 no threshold

In that case only the new flag doesn't have any effect because since no threshold activated the ranging is stopped by the STM32 before going to standby.

If you do not use the EnableTofWakeup=1 the LS will wake up with the RTC.

The Config.UseLowPowerSettings=1 can help to reduce the power.

Using the following configuration:

- set Config.RangeMode=5
- set TimingRepeatInMs=100
- set StandByCountRanging=5
- set StandByTimeoutSec=5
- **set Config.EnableTofWakeup=1**
- **set Config.EnableToFLowPower=1**
- **set Config.UseLowPowerSettings=1**
- savepar

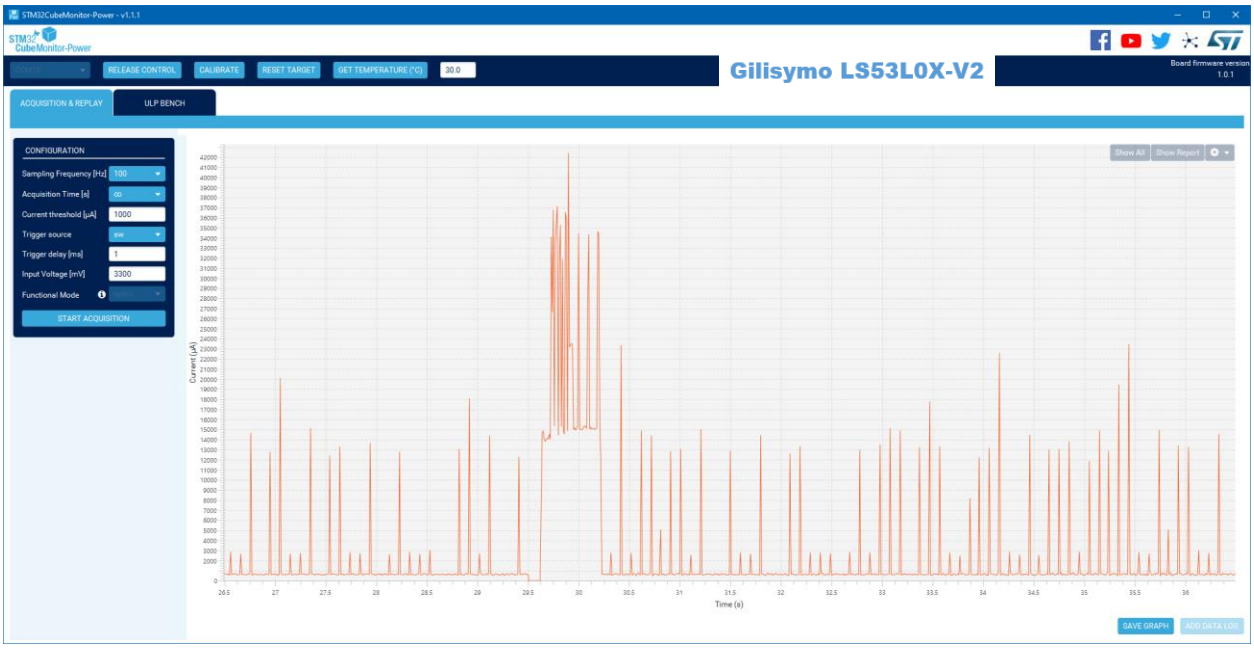


Figure 13: Power Chronogram with EnableTofWakeup=1 EnableToFLowPower=1 UseLowPowerSettings=1 no threshold

In that case **data** from sensor **is very inaccurate**.

8. Standard modes

The following power chronogram corresponds to a configuration in low power modes i.e. from 0 <= RangeMode <= 4, without thresholds.

In standard mode the LS has the possibility to go to standby with the command “standby”. When new parameters are not set, in particular EnableTofWakeup=0 and EnableToFLowPower=0. The behave of the LS is the same as previous FW. When the command is sent, the LS stop the ranging and go to standby. The STM32 will wake up after StandByTimeoutSec seconds.

The new parameter will have in that case a limited effect. In fact, since no threshold is programmed, and no low power mode neither, in case we set both EnableTofWakeup=1 and EnableToFLowPower=1, at the very first ranging from VL53L0X the STM32 wake up and no real gain on power consumption. A threshold programming will no have effect on wake up because the ranging mode is not a low power one.