POWER CONSUMPTION

ble_beacon

Schule der ansprechenden Künste
Andreas Erdmann

1 Power Consumption – ble_beacon

In the first part in chapter 1.2 we measure the power consumption with the current implementation. We'll identify in sections 1.2.14 and 1.2.15.1 that everything is ok but the accelerometer. Thus, in the following chapter 1.3 we will focus on optimizing the accelerometer usage.

Anyway, the different measurements and information we get from different configurations is interesting, so we will keep this information in the document, too, as a reference and starting point for other interested people during their optimization.

1.1 Summary

In this document we show on how we did the power consumption optimization on the Bluetooth low energy (BLE) device based on Nordic Semiconductor NRF52832.

Our first straight forward implementation states a high usage of around 3.6 mA (see section 1.2.6), which gives a life time based on one cell coin battery (CR2032) of ~1.8 days (220mAh/3.6mA * 70% = ~42h), which is far too short. In comparison, the supplier's original firmware of that devices has a consumption of 577 µA, which corresponds to ~11 days (220mAh/577µA * 70% = ~267h), and which is not acceptable, too.

With our optimization we were able to reach an average consumption of 14.4 µA, which gives:

*220mAh / 0,0144mA * 0,7 = 10.694 h = ~1.2 Years*

The **main reason** we identified was the usage of the KX022 accelerometer. In the IIC initialization sequence we must allow the device a 1.2/ODR delay time when transitioning from stand-by to operating mode to allow new settings to load (as stated in the data sheet). Otherwise, the power consumption will stay high. Because there is no delay function in the TWI transaction manager (but you can use blocking functions), we decided to switch to a blocking TWI mode and don't use the transaction manager in a combined blocking and non-blocking mode. See section 1.3.3.1 for the specific results.

A further optimization is to use the RTC INT and go to sleep while waiting for accel data by setting a timer with the appropriate delay, see section 1.4.2. With these measures the power consumption goes down to 15 µA.

Another approach was to use a **nested approach** to first initiate the long running temperature and humidity data request, get the accel data, and then retrieve the temperature/humidity data. Section 1.5.4 shows the results. This gives no noteworthy further improvement.

Finally, **tuning the intervals** between BLE advertising and samples leads us to a result of ~14.4 µA, see section 1.6, which is considered as our new baseline.

I recommend a check against these values for all upcoming releases, e.g. the button introduction, as done in section 1.7.

1.2 Baseline – the starting point

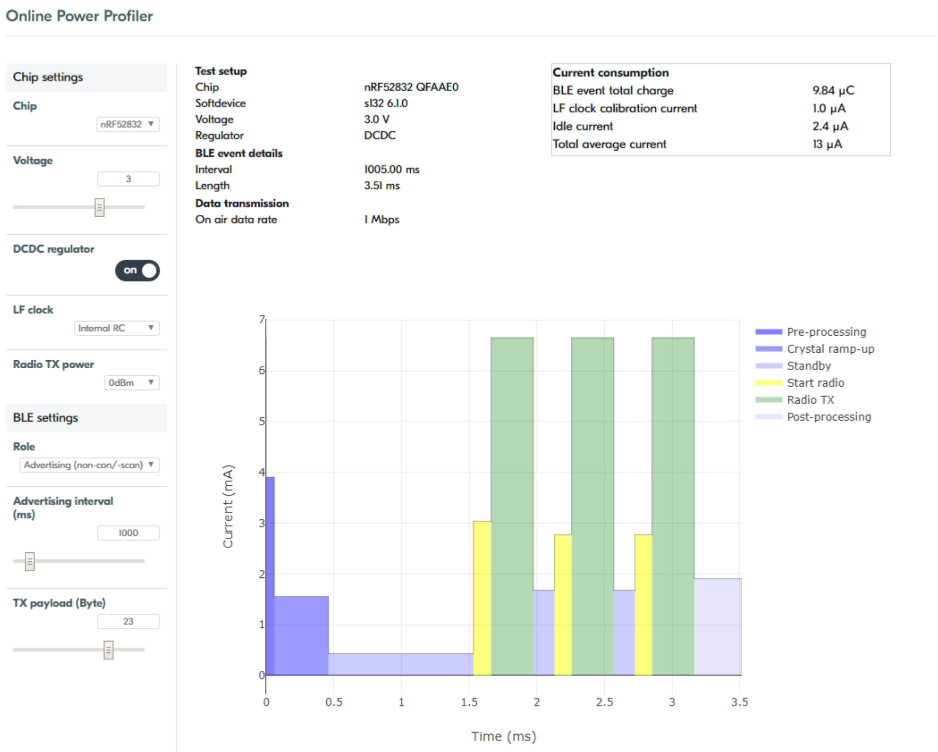
The baseline for the device is measured with only the SoftDevice S132 (Bluetooth 5 Central and Peripheral protocol stack) and no app flashed.

1.2.1 Calculation

From the data sheets for the NRF52832, KX022 (accelerometer) and SHT3 (temperature/humidity sensor) we get the following data which leads to a calculated power consumption.

NRF52 device	
Idle current	2.4 µA
Total average current	13 µA

The NRF52 power consumption is based on this calculator ([Link]()) with this results:



KX022 device	
High power mode	145 µA@2.5 V
Low power mode	10 µA
Standby	0.9 µA

SHT3 device	
Idle state	0.2 µA (max 2 µA)
Average	2 µA (while measuring w/lowest repeatability + single shot, 1 Hz)

Overall, we thus can expect a power consumption in idle mode as stated in the following table:

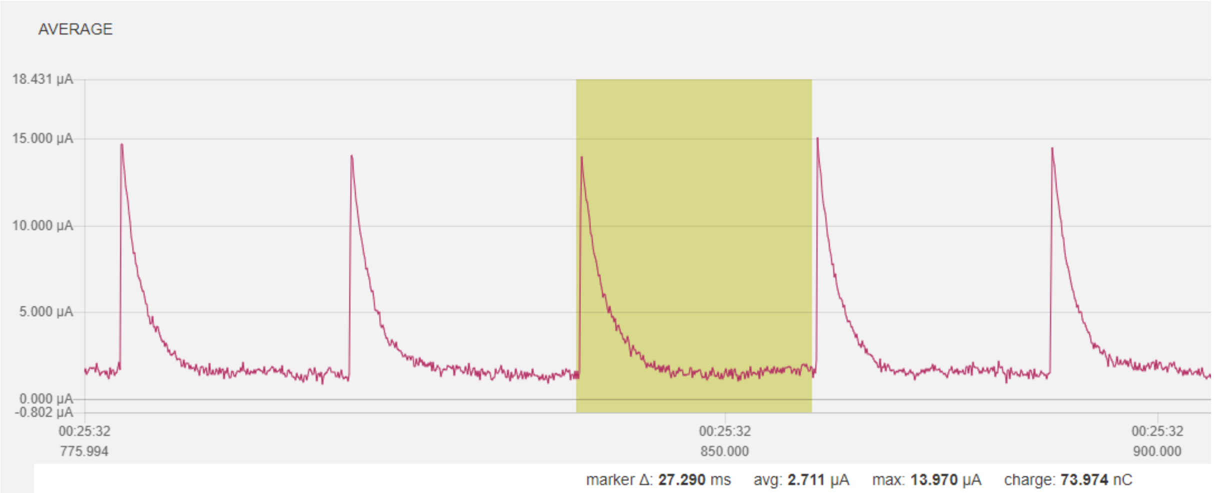
Overall Idle Current	
NRF52 (Idle current)	2.4 µA
KX022 (standby)	0.9 µA
SHT3 (Idle state)	0.2 µA (max. 2 µA)
SUM	**3.5 µA**

With the active configuration stated in section 1.6 we can expect an average current as summarized in the following table:

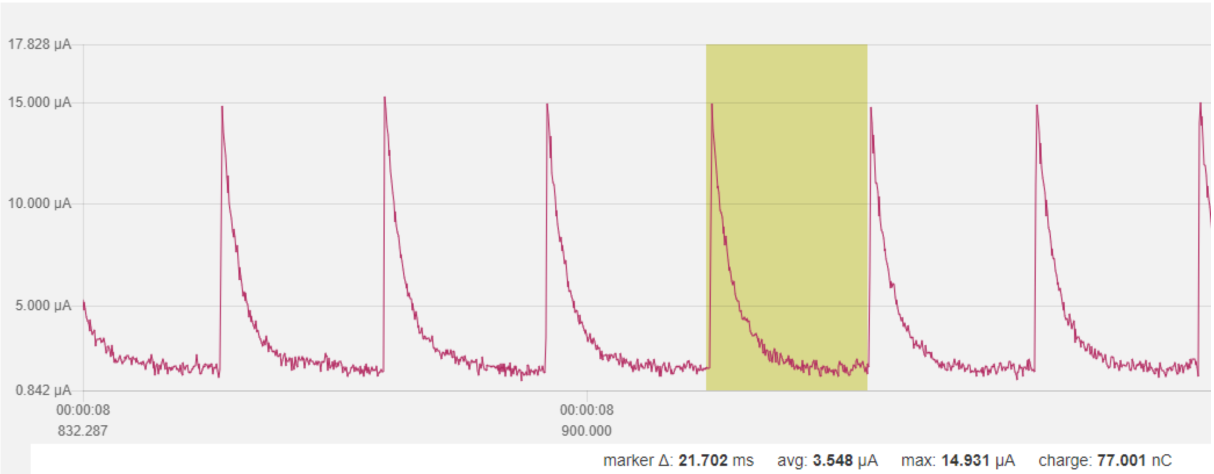
Overall Current	
NRF52 (Total average current)	13 µA * 60 s
KX022 (each 15 sec = 4/min)	+ 320 µA (= 10 µA * 8 ms * 4)
SHT3 (each 15 sec = 4/min)	+ 96 µA (= 2 µA * 12 ms * 4)
SAADC (each 60 sec = 1/min)	+ Xxx (= 10 µA * x ms)
SUM	**~13-14 µA (average current)**

With all the measurements and optimization, we reach a value which is close to this theoretical bottom line. Done!

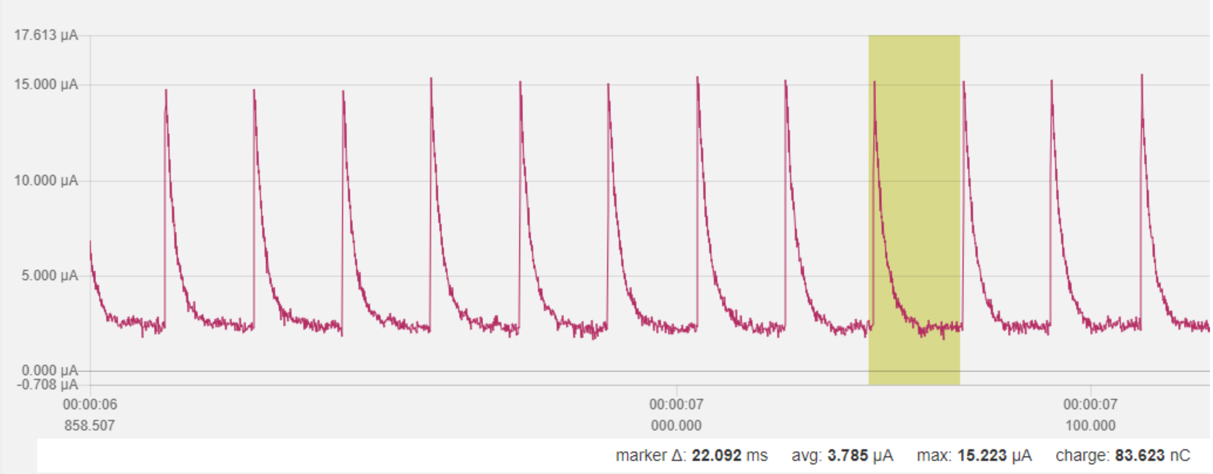
1.2.2 SoftDevice flashed, no app flashed



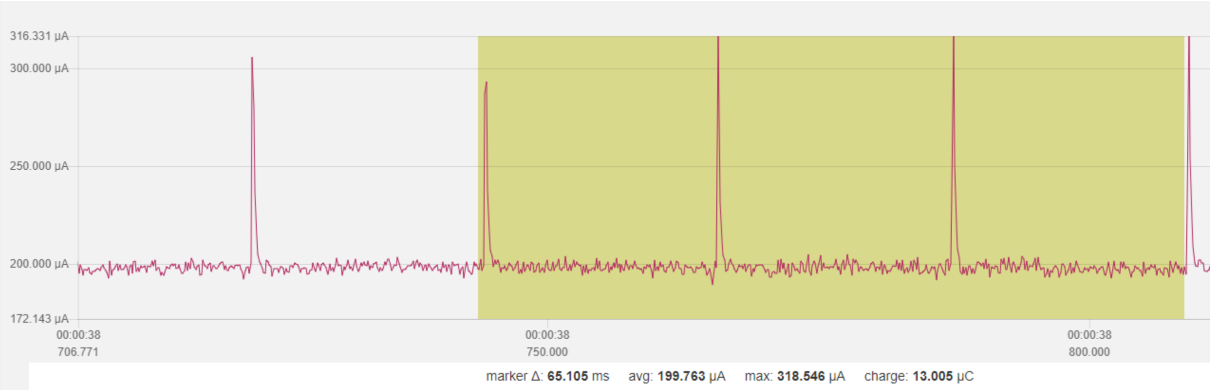
1.2.3 SoftDevice, just go to idle mode



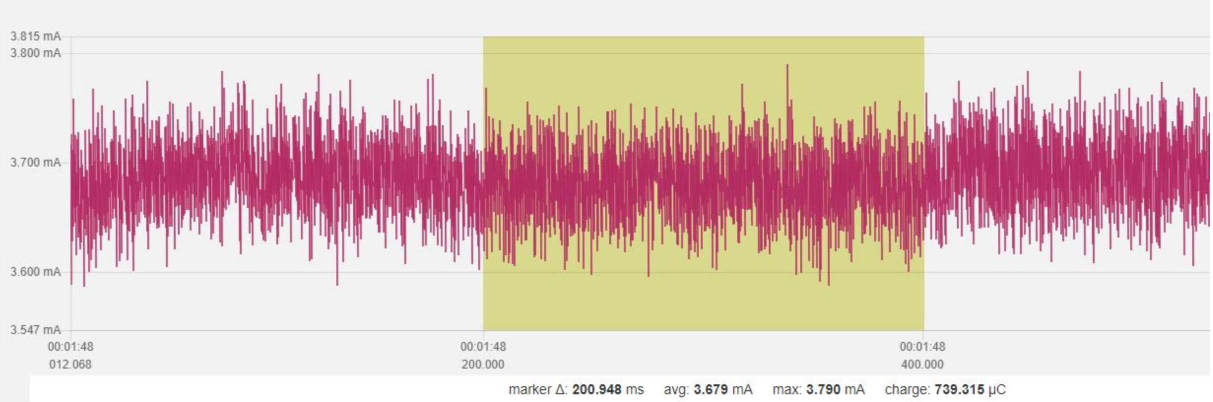
1.2.4 SoftDevice, init bsp (led off), and just go to idle mode



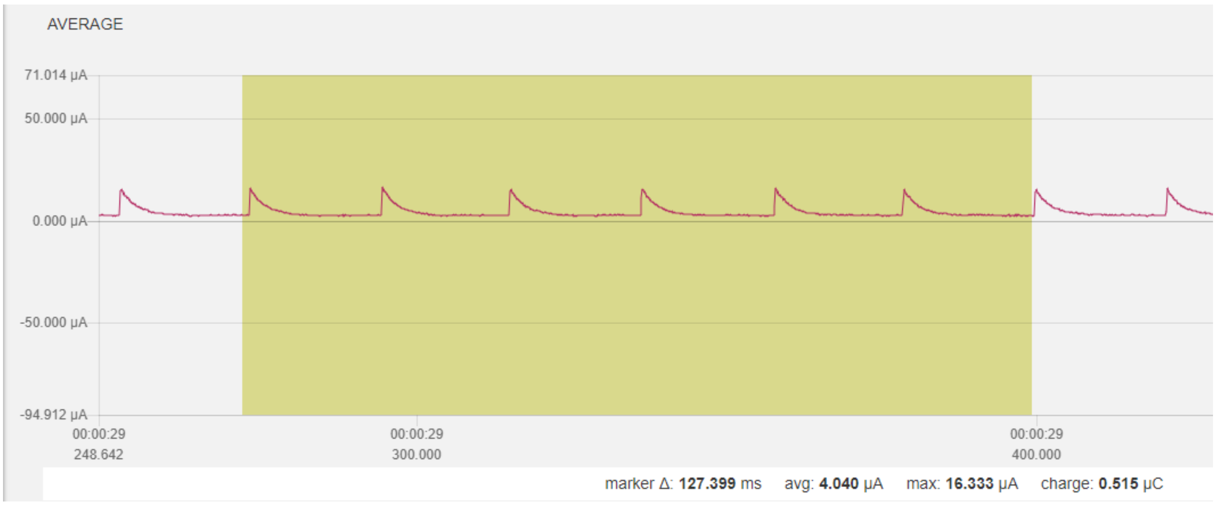
1.2.5 SoftDevice, init bsp (with one led on), and just go to idle mode



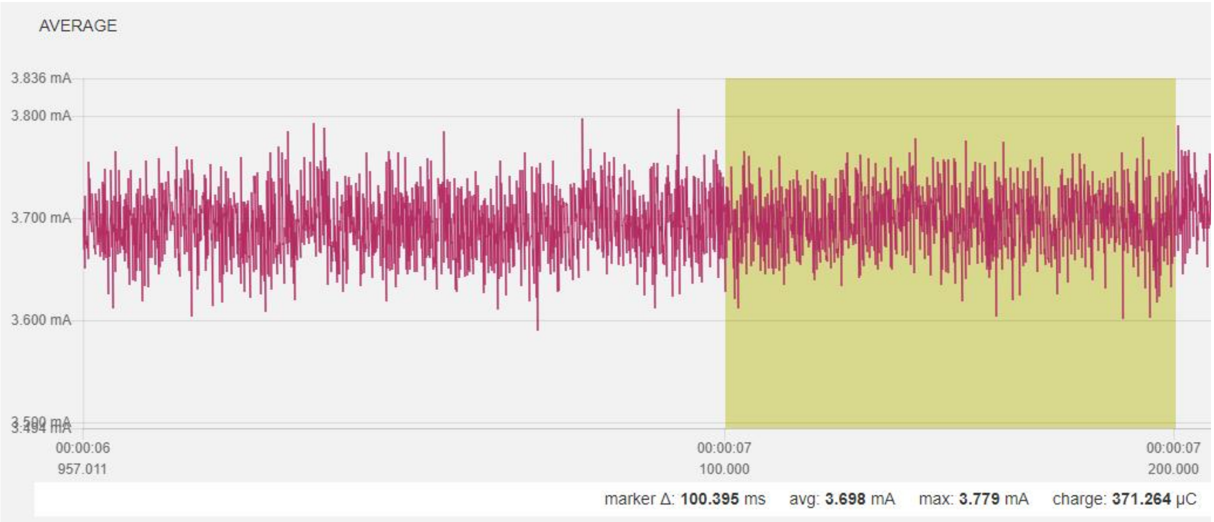
1.2.6 SoftDevice, init bsp (led off) + twi_config + (both) sensor_init



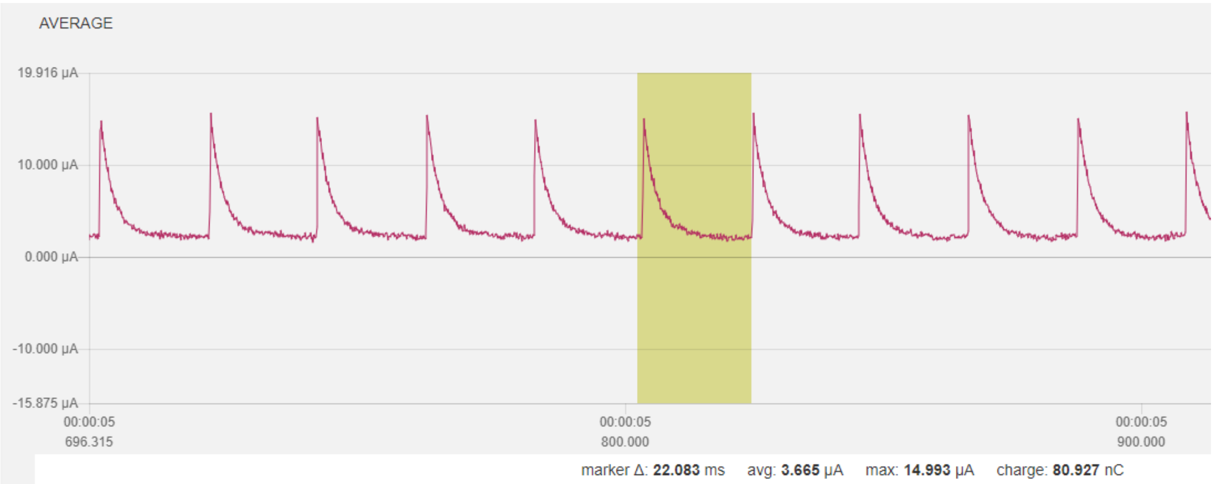
1.2.7 SoftDevice, init bsp (led off) + twi_config + no sensor_init



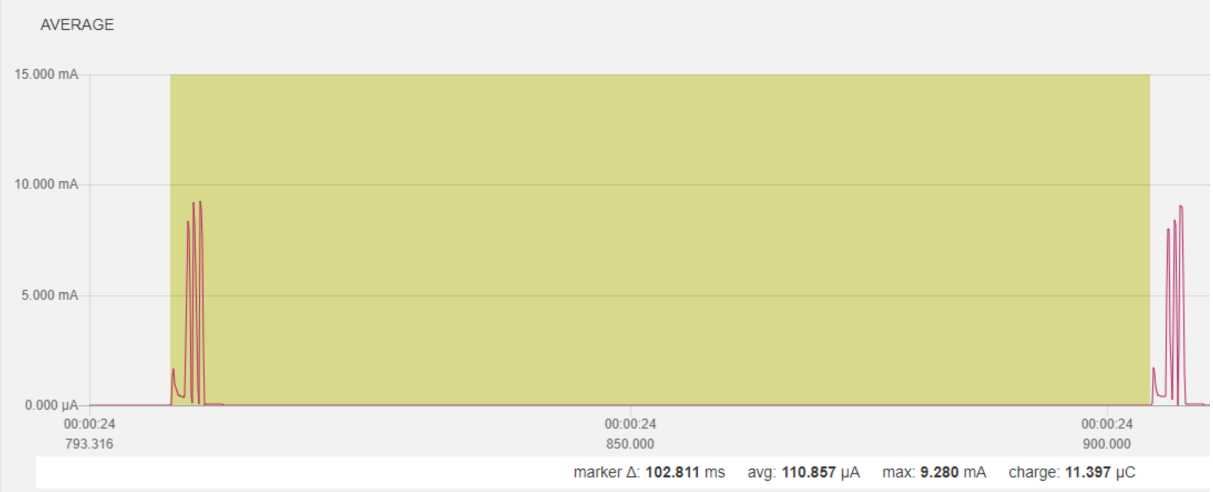
1.2.8 SoftDevice, init bsp (led off) + twi_config + only kx022 init



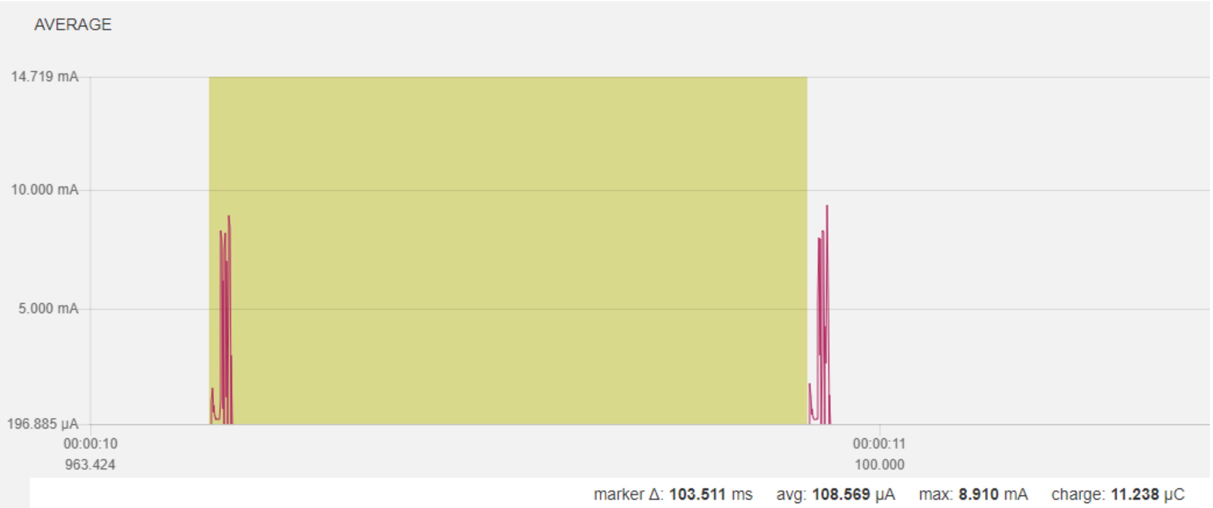
1.2.9 SoftDevice, init bsp (led off) + twi_config + only SHT3 init



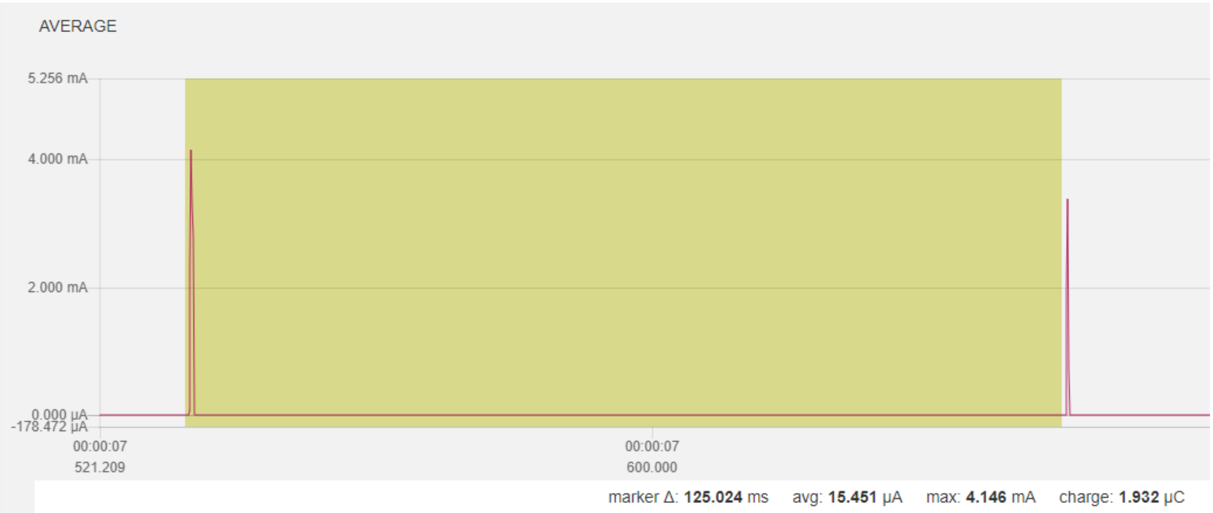
1.2.10 SoftDevice, init bsp (led off) + twi_config + SHT3 but no KX022, no sensor data + BLE adv



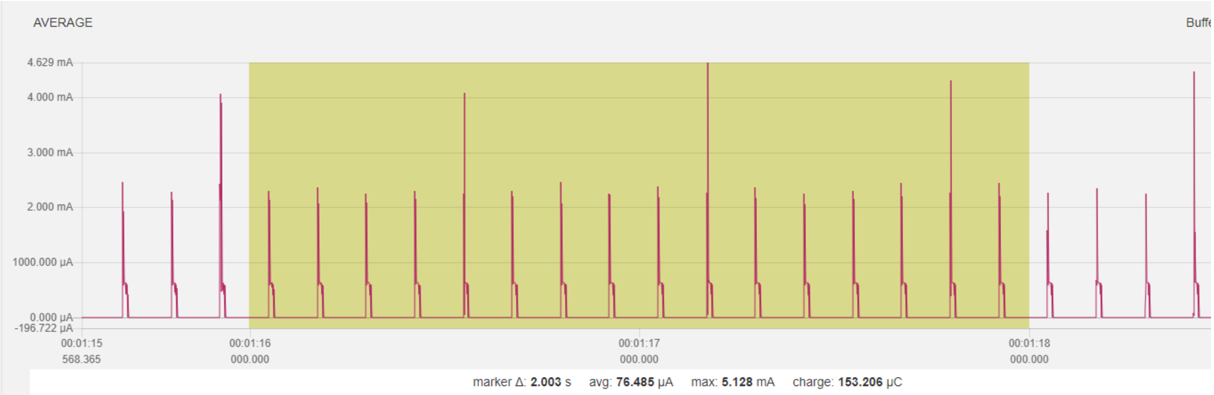
1.2.11 SoftDevice, init bsp (led off) + twi_config + SHT3 but no KX022, no sensor data + BLE adv + SAADC measurement



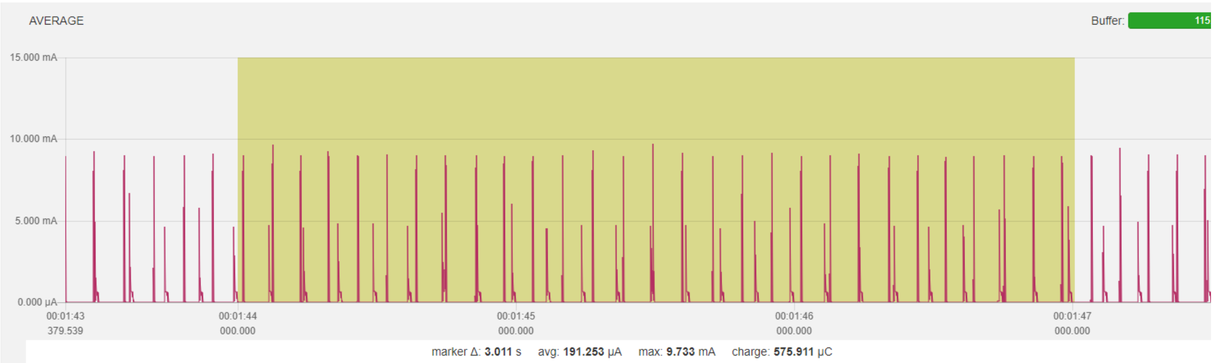
1.2.12 SoftDevice, init bsp (led off) + twi_config + SHT3 but no KX022, no sensor data + without BLE init/adv + SAADC measurement



1.2.13 SoftDevice, init bsp (led off) + twi_config + SHT3 but no KX022, SHT3 measurement + without BLE init/adv + SAADC measurement

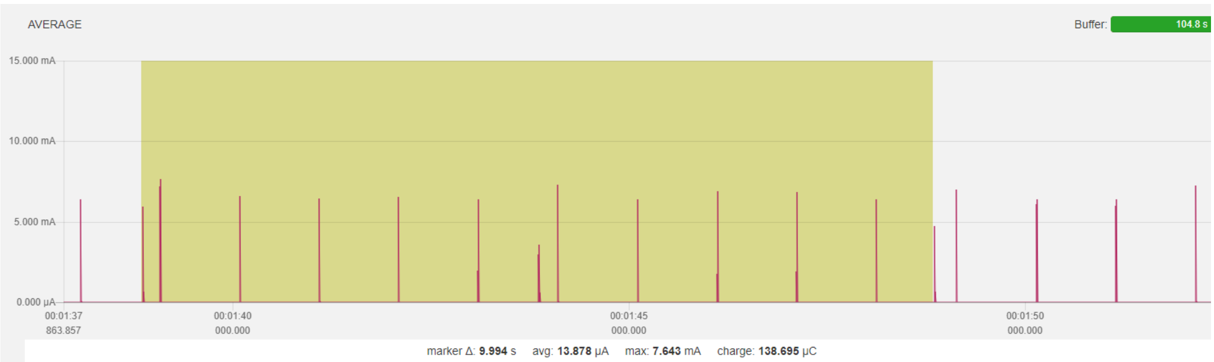


1.2.14 all but KX022 measurement, 1/8 data acquisition for SHT and SAADC

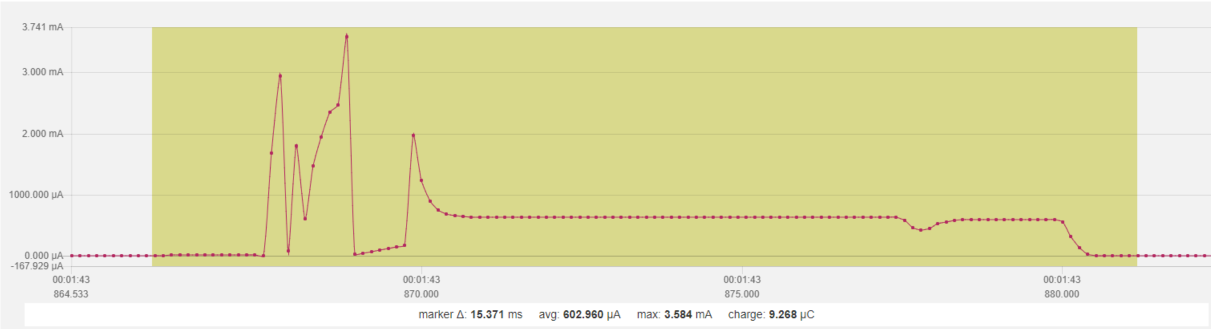


1.2.15 on top of 1.2.14 use different parameter

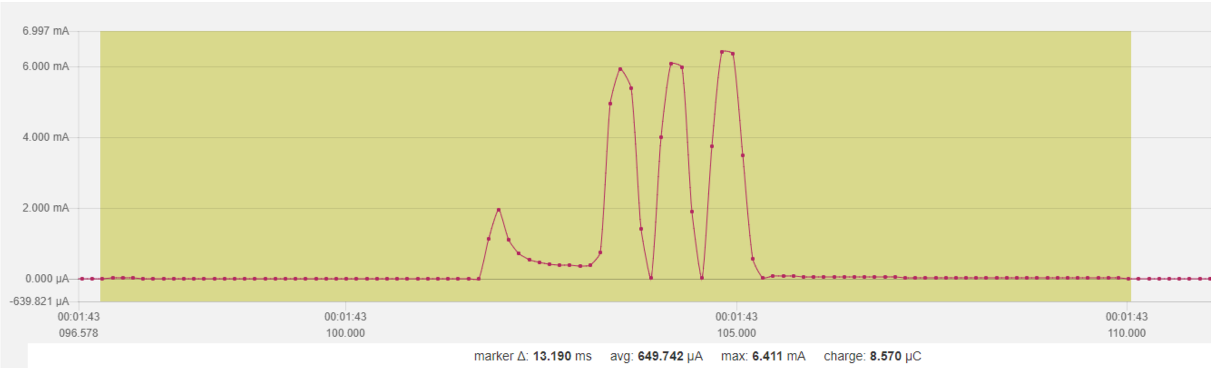
- reduce transmit power to 0dBm from +4dBm
- adv int to 1 sec from 1/10 sec
- SHT update int to 5 sec from 1/8 sec
- SADC update int to 10 sec from 1/8 sec



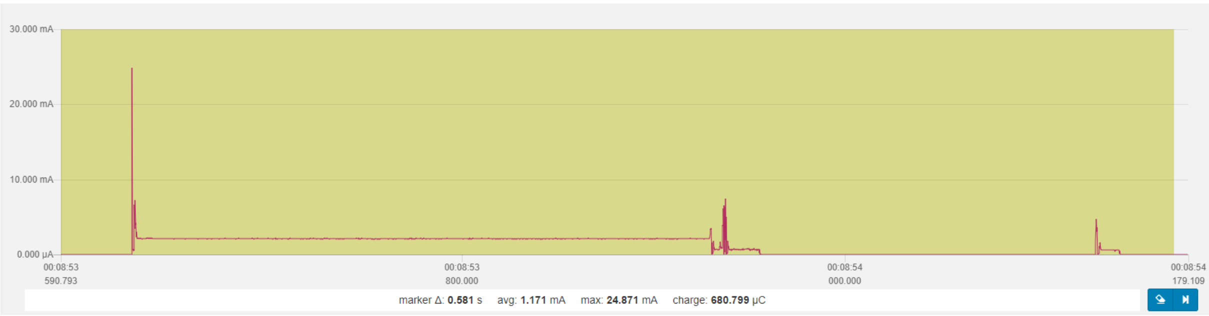
1.2.15.1 Sensor update (all 5 sec)



1.2.15.2 Adv (all 1 sec)



1.2.15.3 Startup

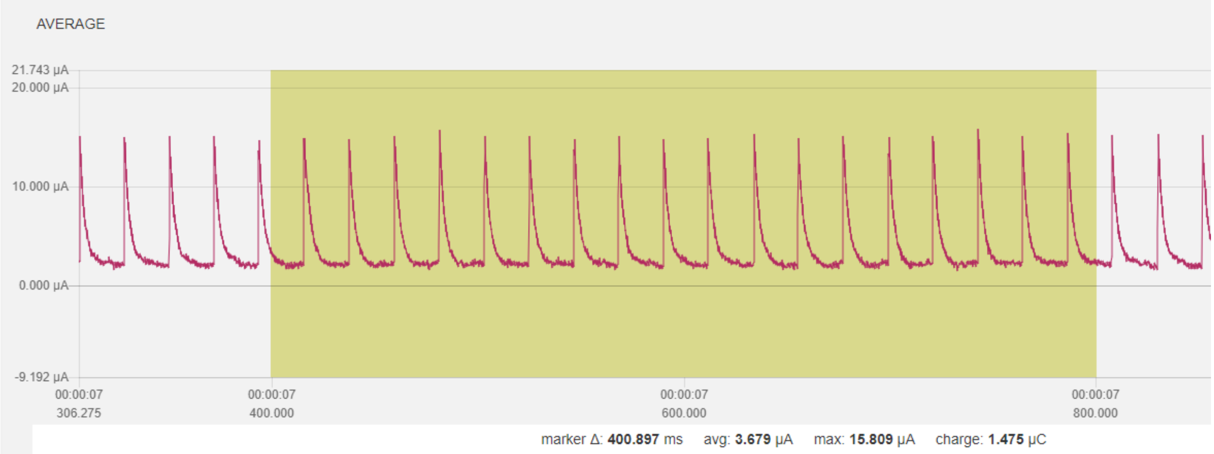


- Peaks
  - Power on peak
  - First adv
  - First sensor acquisition

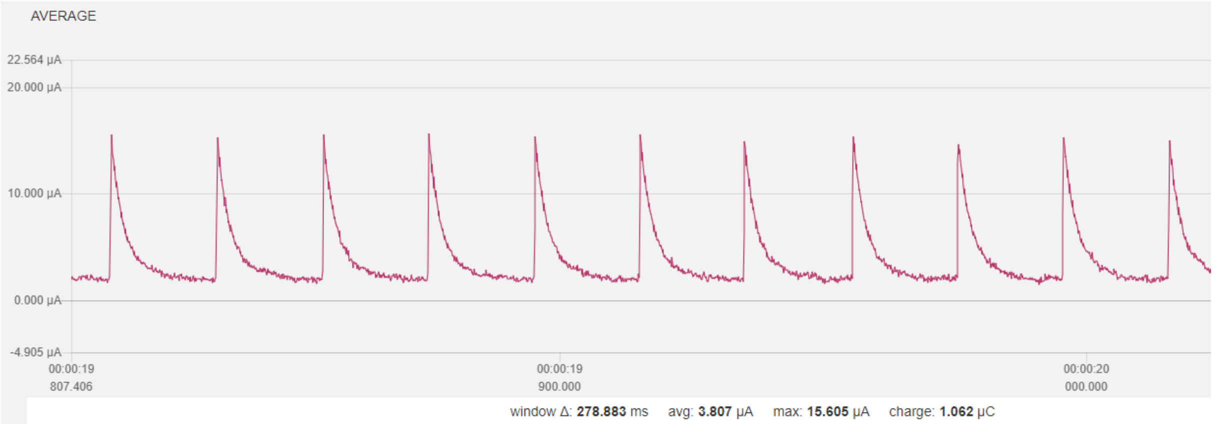
1.3 Power Optimization KX022 Accelerometer

As we saw in the previous chapter, we need to focus on optimizing the KX022 accelerometer power consumption.

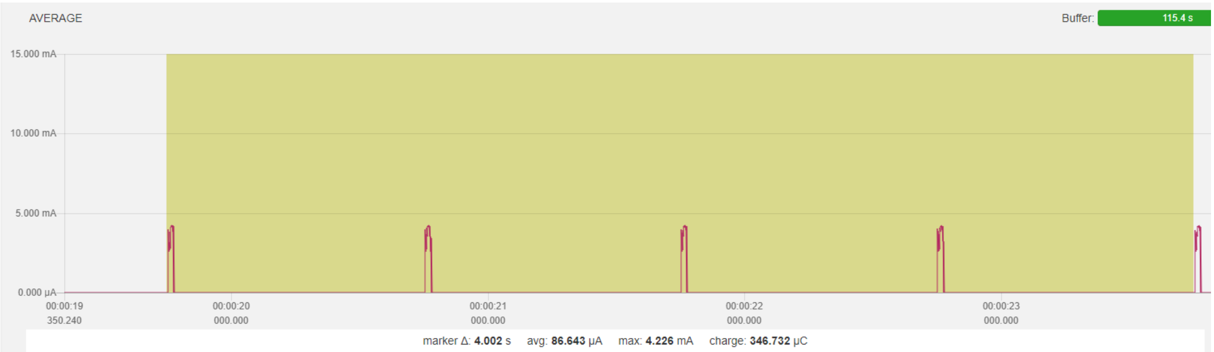
1.3.1 Baseline, no BLE, no sensor init



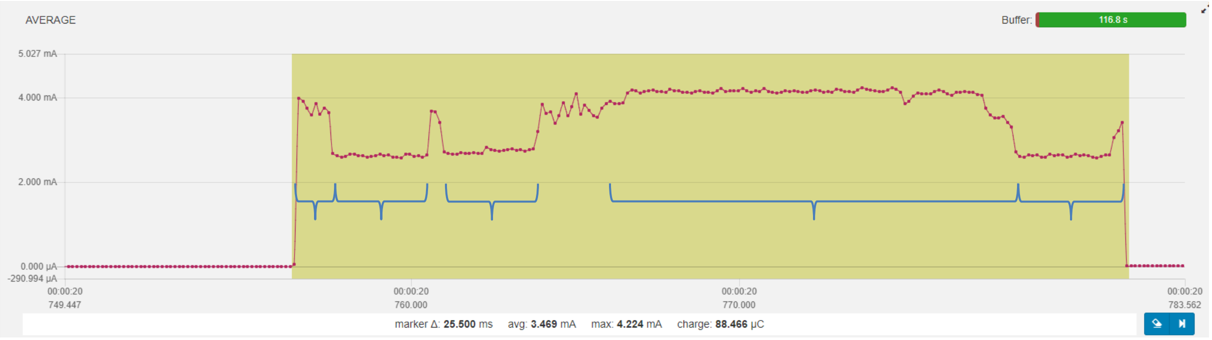
1.3.2 Change to TWI without transaction manager, SHT3 init and KX022 init to standby



1.3.3 With KX022 and SHT3 “one shot” measurement, 1 Hz

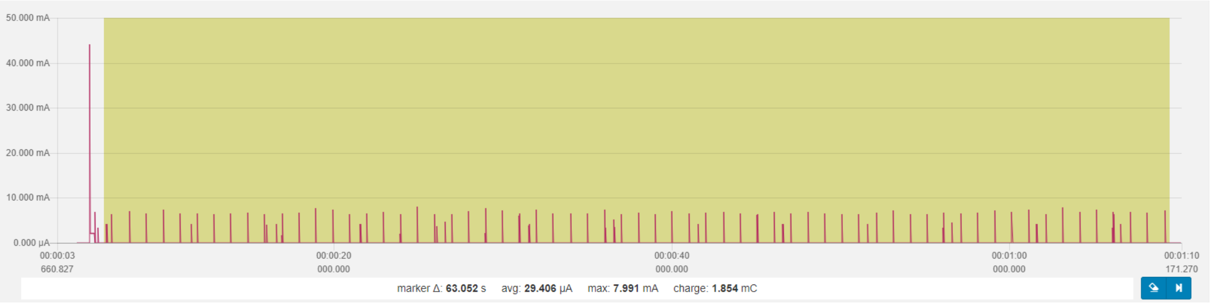


1.3.3.1 One measurement

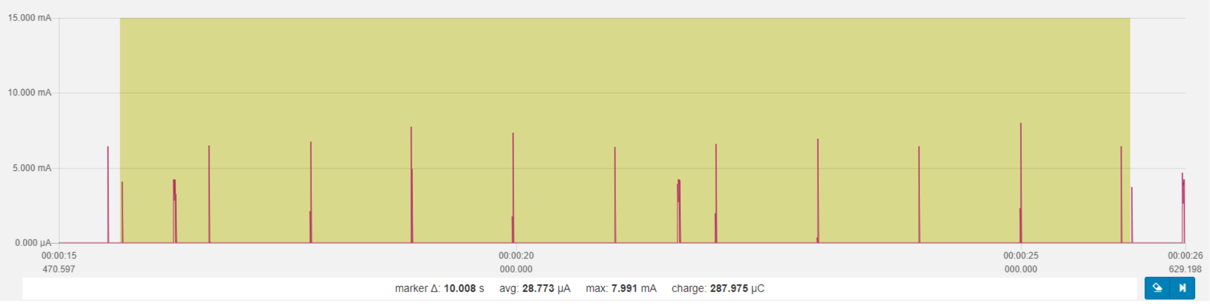


Init KX022, Standby	1,2ms
wait 1.2/ODR	3ms
set to operate	0,5ms
wait 1.2/ODR for value	3ms
read accel values	…
SUM	~8ms
Set SHT3 to SHT3_MEAS_HIGHREP_STRETCH	
wait clock stretch	12,5ms
read temperature and humidity	2,8ms
SUM	~15ms
Process data and sleep again…	
Overall cycle	25ms, avg. power consumption 3,5mA, idle < 4uA

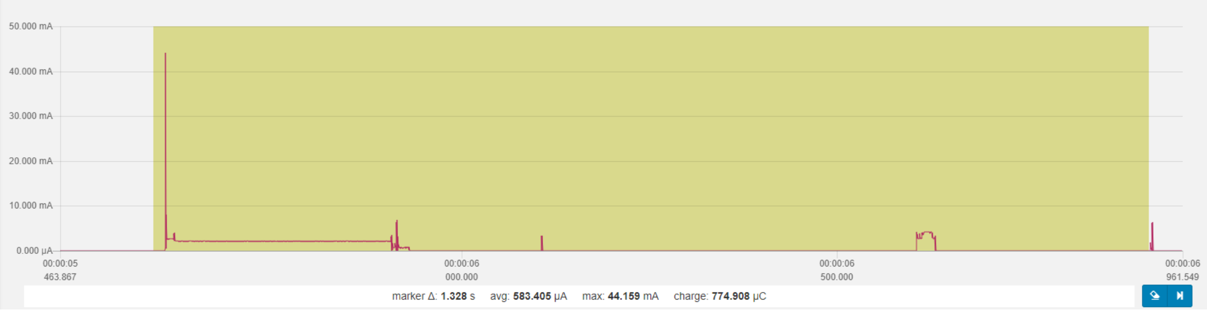
1.3.4 Overall



1.3.4.1 One 10sec cycle



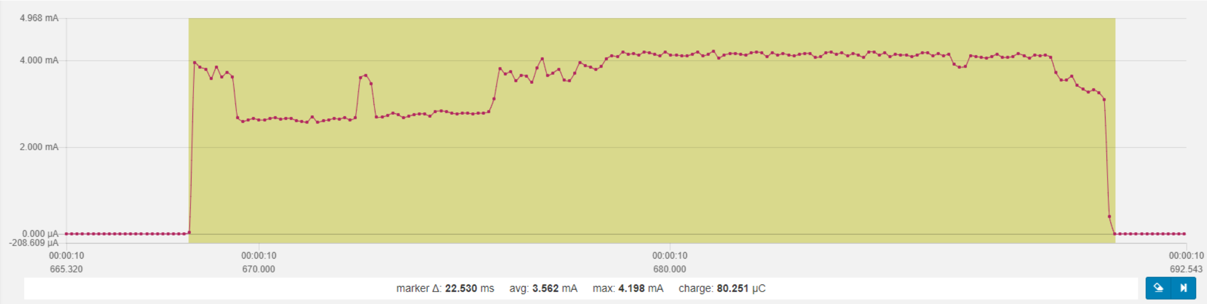
1.3.5 Cycle



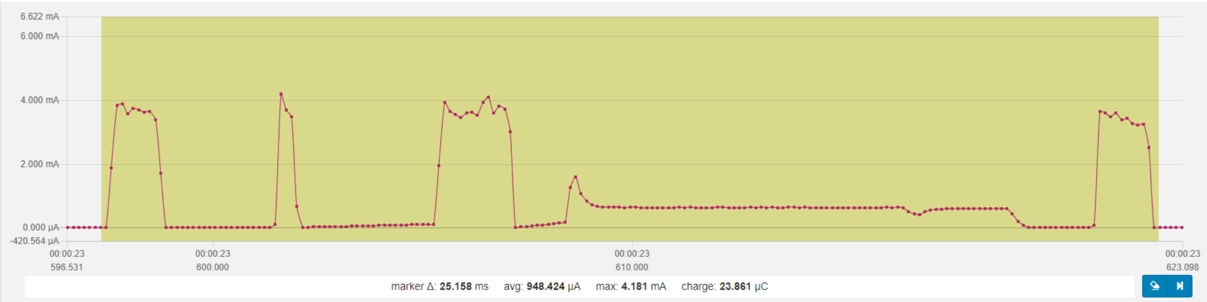
BLE 0 dBm, adv. interval	1s
sensor (SHT3 and KX022) interval	5s
SAADC (battery level) interval	10s
overall power consumption	~30uA (28,77uA)
idle power consumption	3,5uA

1.4 Use RTC INT for while waiting for accel data

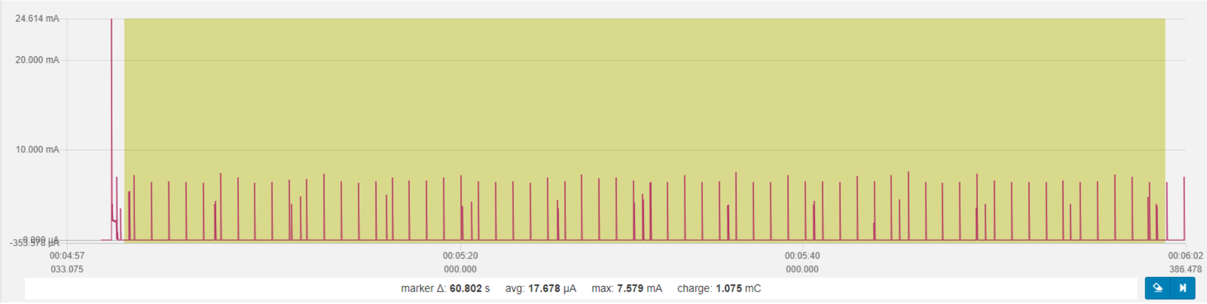
1.4.1 Baseline



1.4.2 Use RTC counter (freq 1/256) for KX022 "put to operation", "wait for accel data", and during SHT3 temp/hum measurement (w/max. 15ms time)



1.4.3 one minute, analog to 1.3.4



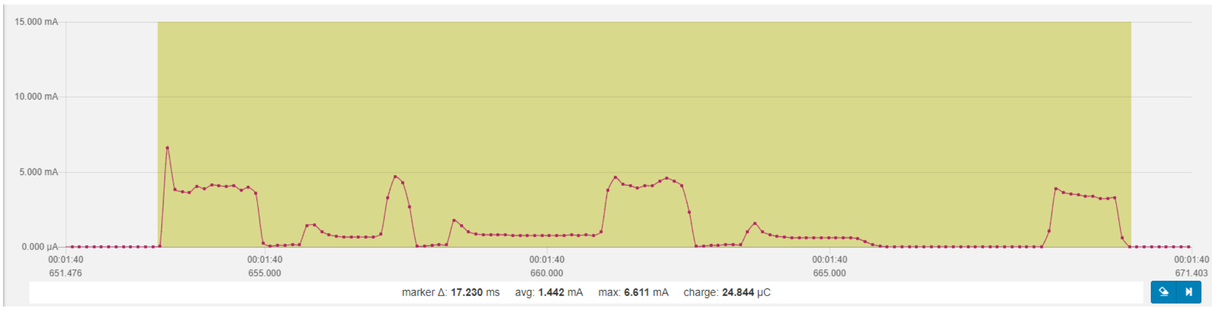
Optimization summary (1.3.4→1.4.3)

BLE 0 dBm, adv. interval	1s
sensor (SHT3 and KX022) interval	5s
SAADC (battery level) interval	10s

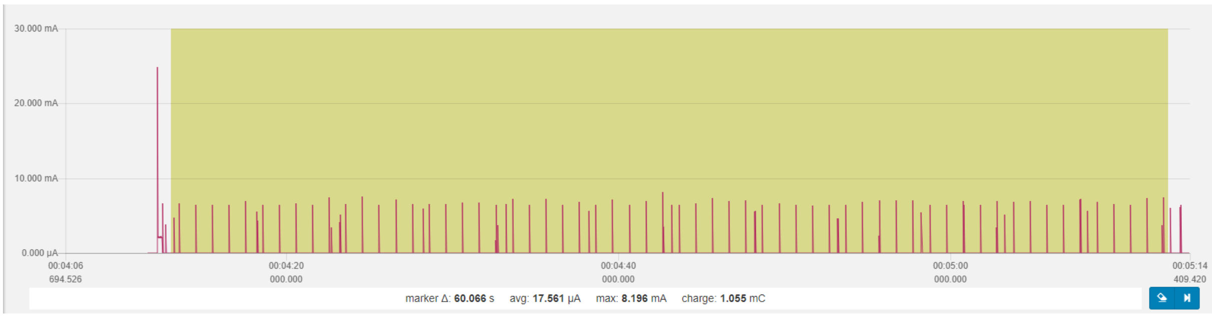
overall power consumption	~30uA (28,77uA)	→ 17,68uA
idle power consumption	3,5uA	→ 3,5 uA

1.5 Further optimization

1.5.1 Using nested approach: start long running SHT3 first, complete KX022 tasks and read SHT3 values KX022: ODR 1600 -> delay time 3ms

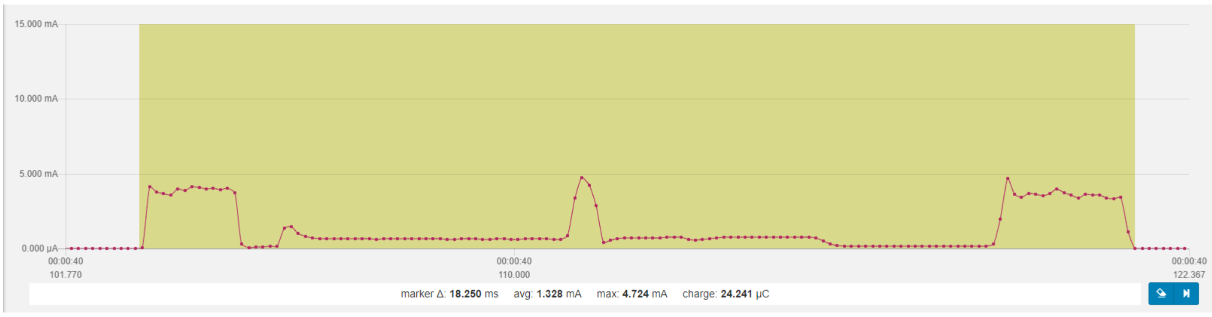


1.5.2 one minute, analog to 1.4.3

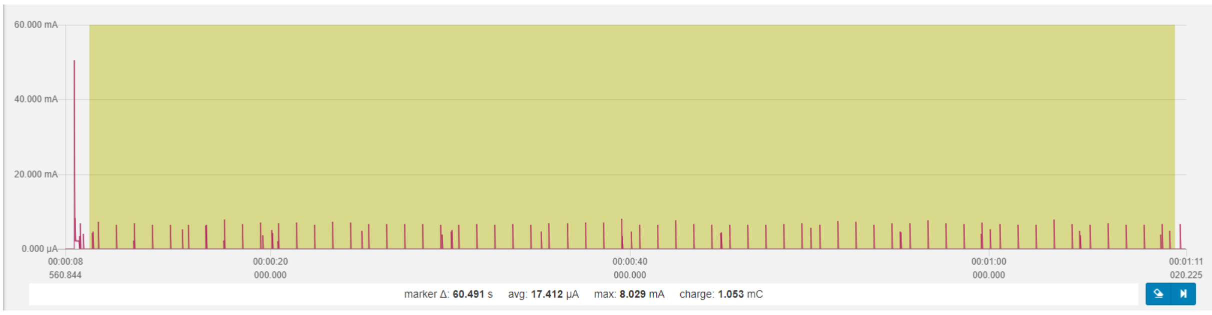


thus, no real further improvement

1.5.3 KX022: ODR 200 -> delay time 7ms

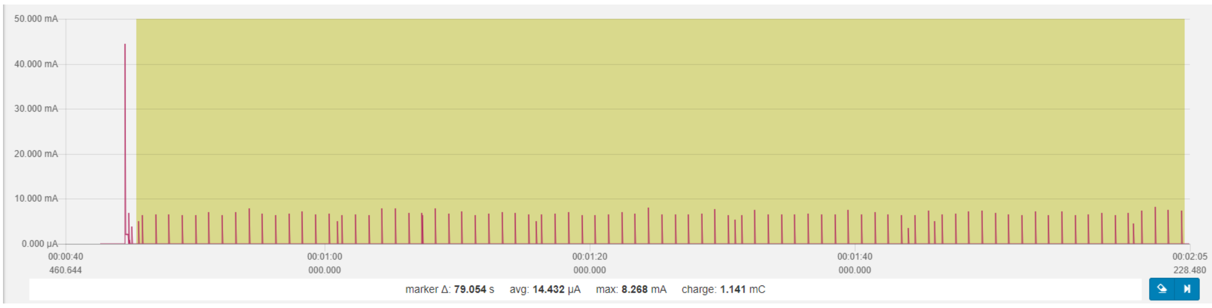


1.5.4 one minute, analog to 1.4.3



thus, no real further improvement

1.6 Longer intervals between adv and samples

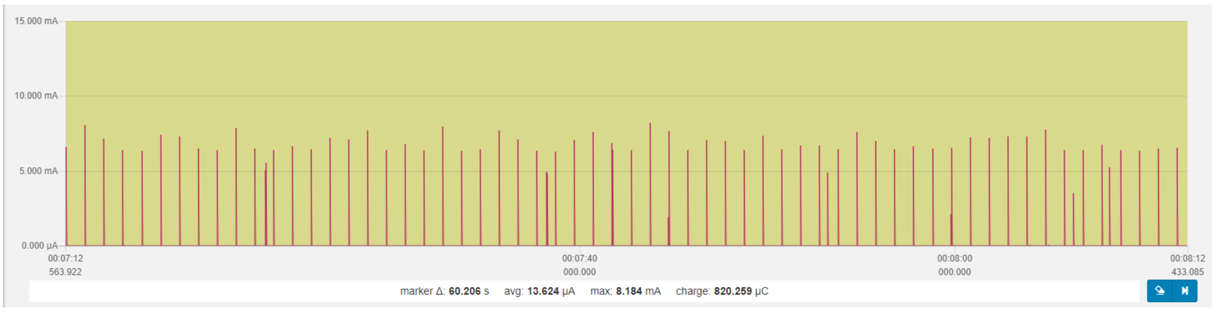


BLE 0 dBm, adv. interval	1s
sensor (SHT3 and KX022) interval	15s
SAADC (battery level) interval	60s
overall power consumption	14,4uA
idle power consumption	3,6 uA

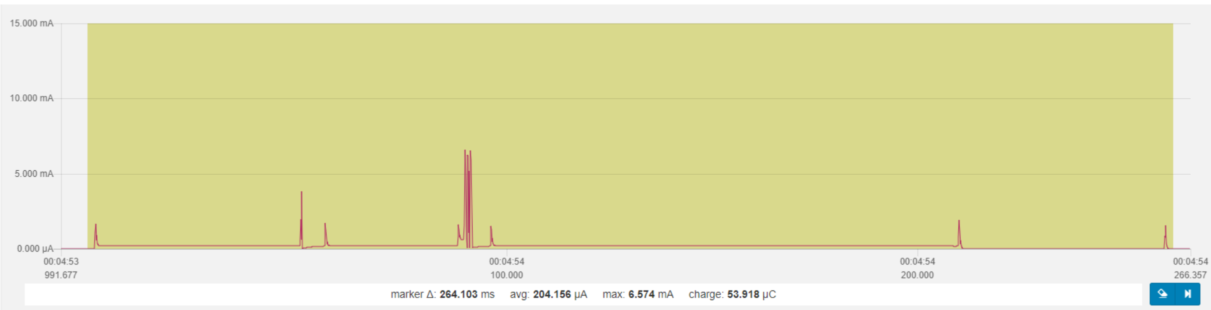
*220mAh / 0,0144mA * 0,7 = 10.694 h = ~1.2 Years*

(> 1 Year = 365 * 24h = 8760h; CR2032 = 220 mAh)

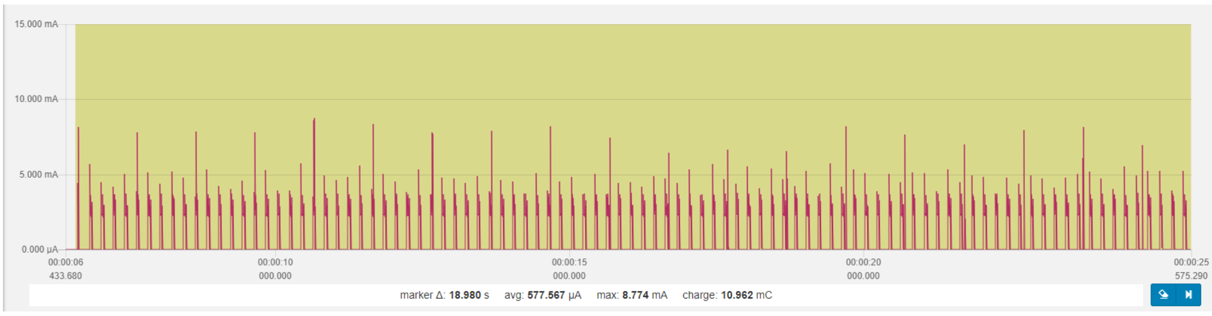
1.7 Button functionality introduced



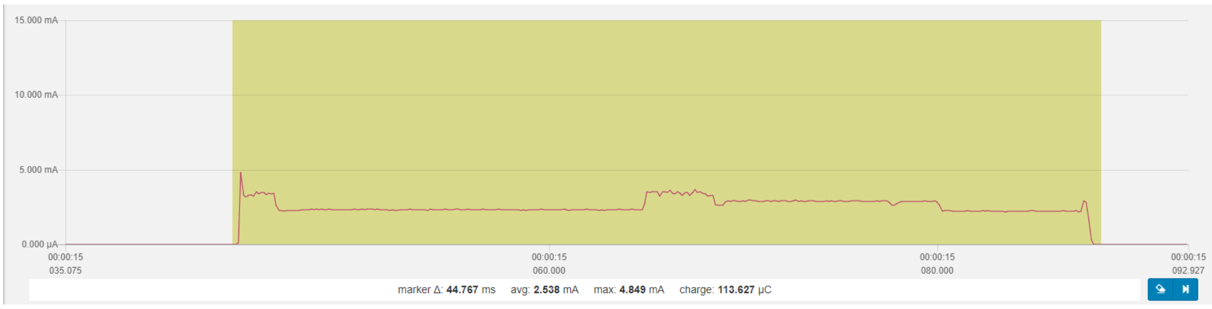
Single button press



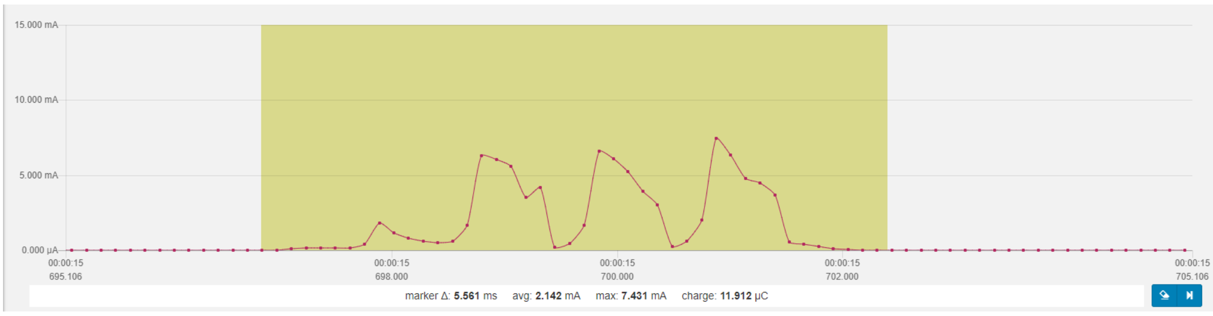
2 Original Beacon Firmware (for comparison)



Frequent spikes (5 Hz)



Larger but rarer spikes (1 Hz)



3 Used configuration

Component	Remark
BLE Beacon with KX022, SHT3, Button, LED	Product Link from Alibaba
Nordic Semiconductor NRF52832	
nRF5 SDK	Version: 15.2.0
SoftDevice S132	S132 SoftDevice v6.1.0
J-LINK EDU MINI	Segger Microcontroller Systems
Nordic Power Profiler Kit	Nordic Semiconductor
nRF52840 Dongle	Nordic Semiconductor
KEIL MDK-Lite	Version: 5.27.0.0
Github repository	https://github.com/an-erd/ble_beacon

4 Actual Power Consumption

4.1 13.04.2019 – 03.06.2019

4.2 13.04.2019 – 03.06.2019

- 19.05.2019: hail, sensor temperature down to ~0°C, and humidity up to ~100%