**NOTES FROM DISCUSSION:**

* Look into whats feasible in the advertisement packet : imagine that you can send a short version of an ID together with the signal strength of the two strongest ones and send it to the backend 🡺 Than he thinks that is enough to make the chair lay-out again
* MW thinks that continuous advertisement and **random** (to make sure you are not listening at the same time as the chair which is next to you) listening of 30 seconds.
* To find the power consumption 🡺 if you know the power consumption of listening and the power consumption of advertising, than you can calculate the power consumption for a determined time frame. That’s how you can determine what battery, energy harvesting… is possible and what not. 🡺 if I know this, I can determine if rf harvesting is sufficient enough or solar is better.

**INFO THINGY:**

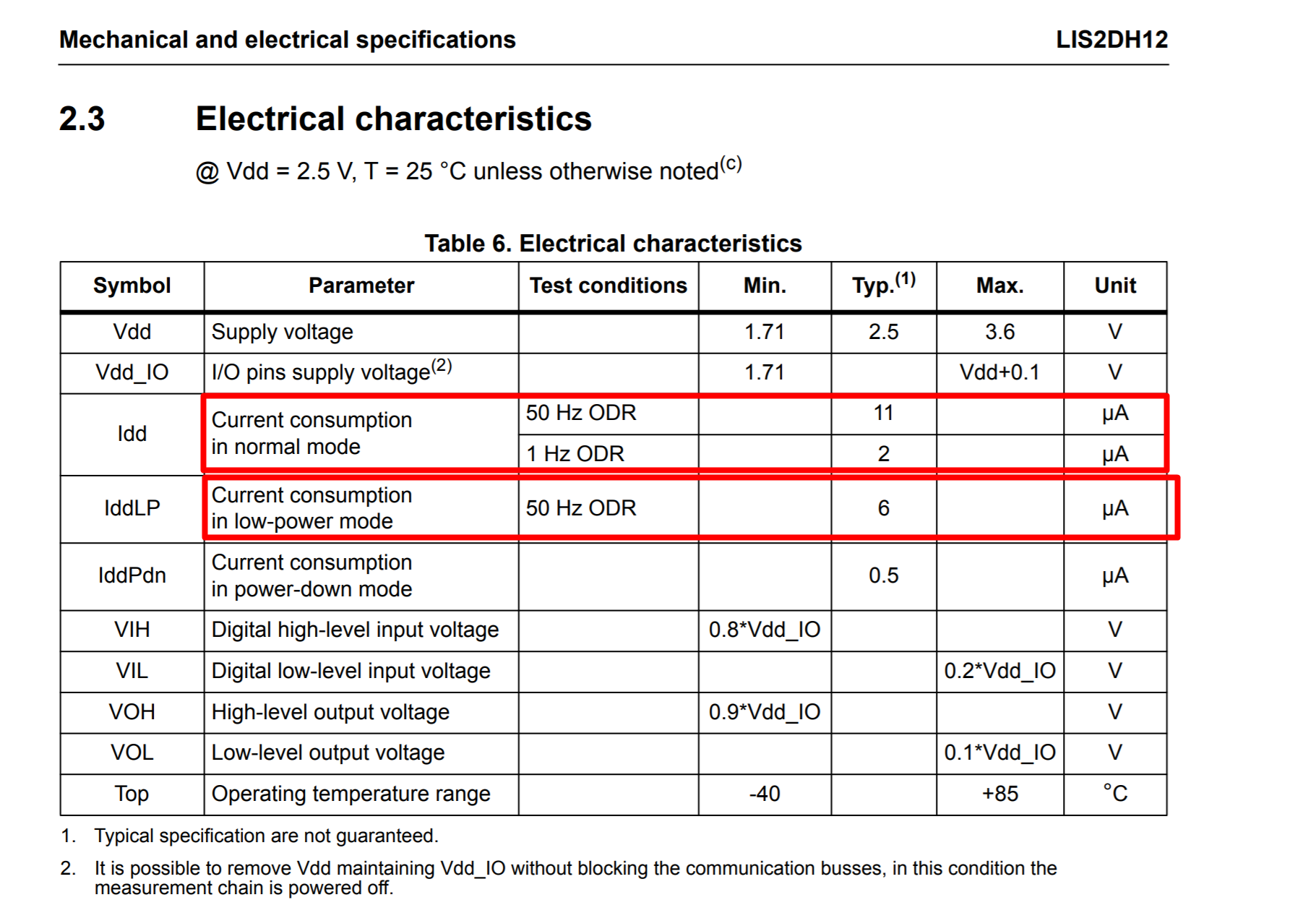
(<https://infocenter.nordicsemi.com/pdf/Thingy52_UG_v1.2.pdf>) 🡺 this gives a lot of info about the the thingy:

* The battery that it uses can be recharged by micro usb
* The battery has a capacity of 1440 mAh 🡺 this means that if 1440 mA is consumed every moment, then it can last an hour
* The max charge current is set to 0.5 C (a C rate of 1C is 1440mAh but the max charge current is set to 0.5C so that means that only half of 1440 mA is allowed = 720 mA to charge the battery. This results in that the battery will last be recharged in 2 hours)
* Will advertise the whole time
* Als scannen 10 seconden duurt en thingy advertised om de second, kunnen er maar 10 thingy’s zijn.

The only things we are going to use from the thingy are: CPU, BLE, Accelerometer, clocks.

**POWER CONSUMPTION ESTIMATION:**

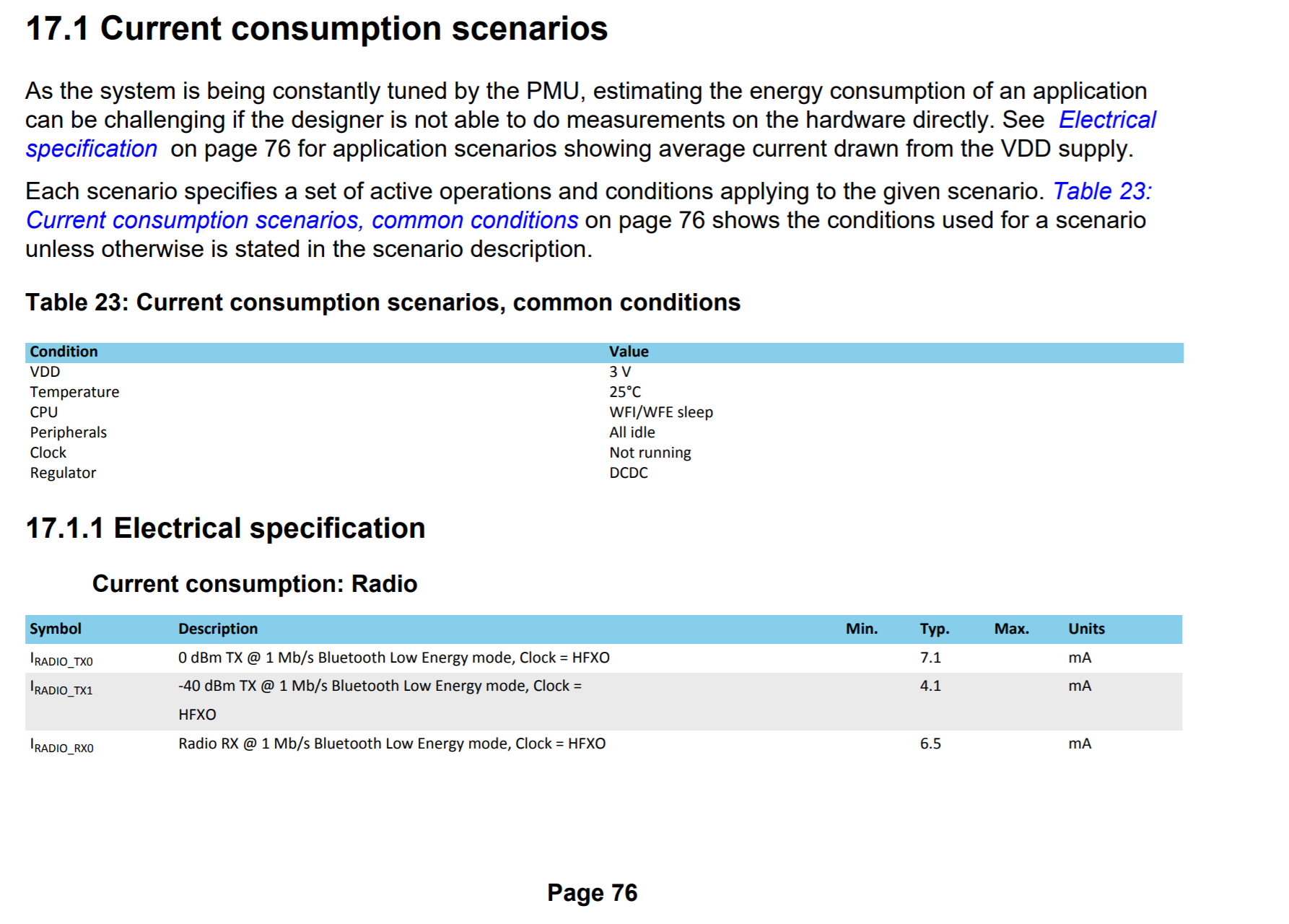
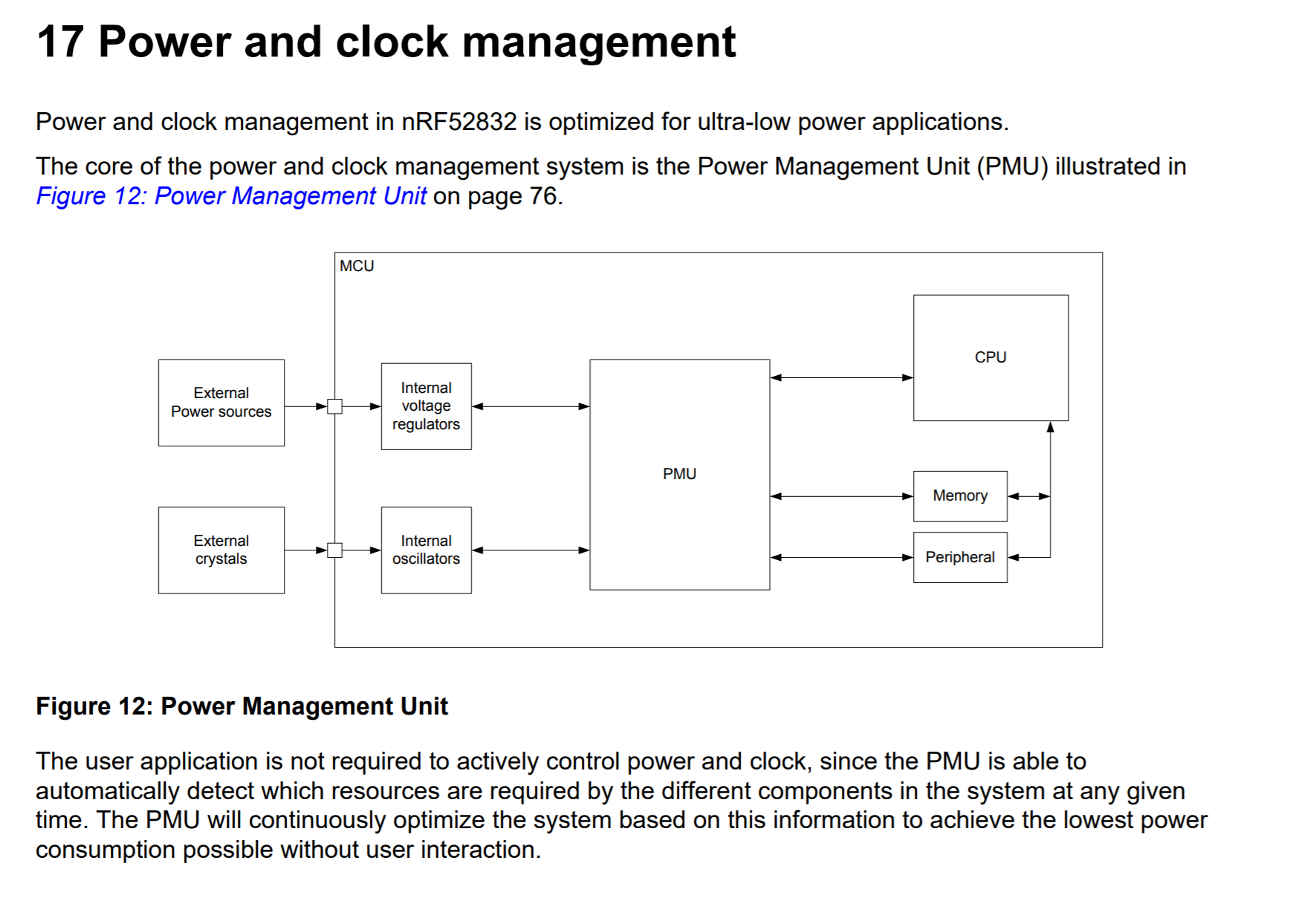
Accelerometer (<https://www.st.com/resource/en/datasheet/lis2dh12.pdf#page=12&zoom=100,81,117>) :

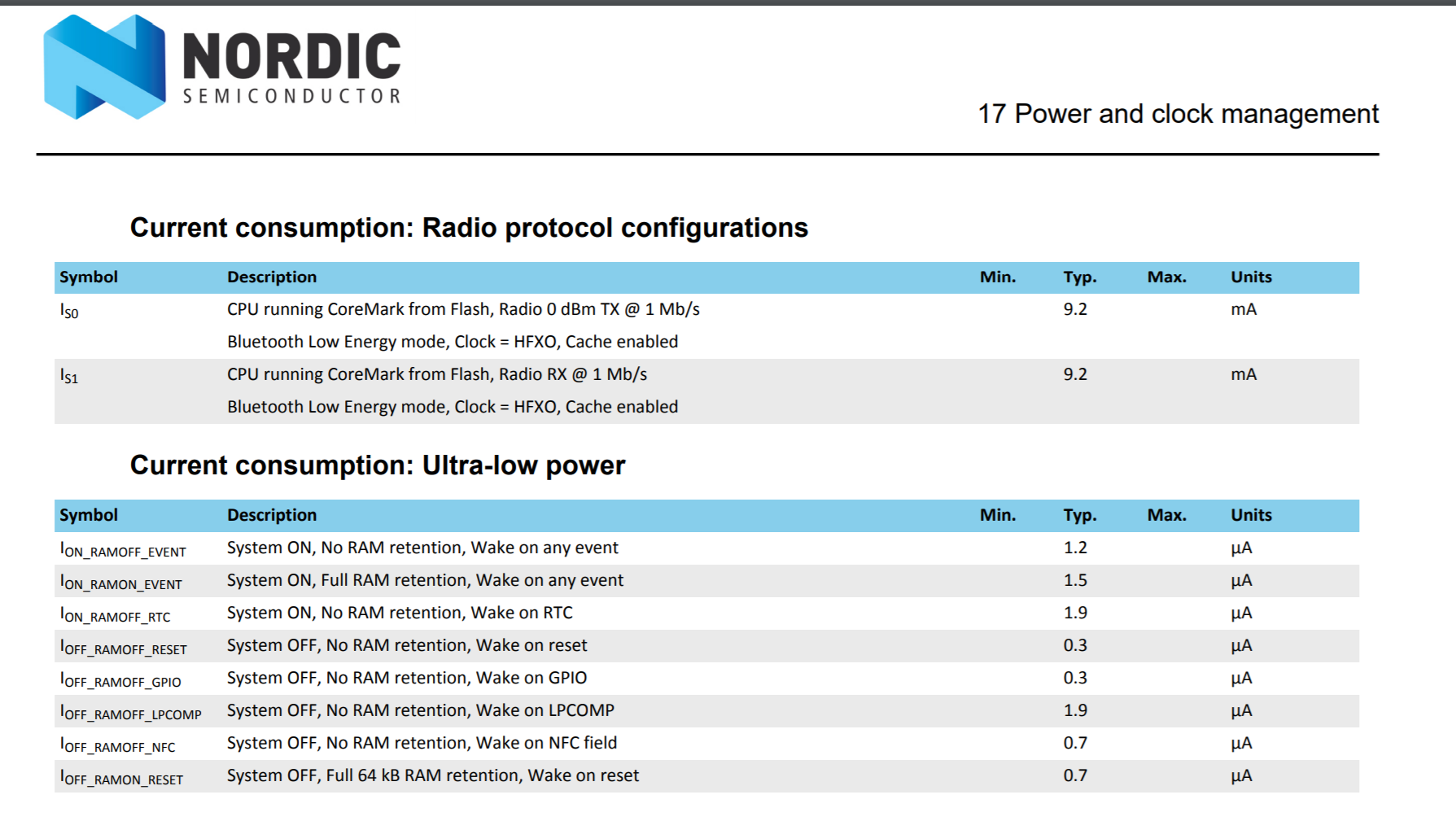


Accelerometer consumes 6 µA for Low-power mode (🡺6µAh) and 11 µA (🡺 11µAh) for Normal mode (so when it is sensing a different position I think).

CPU/BLE-radio (nrf52832)

Datasheet nrf52832 : <https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct_nrf52%2Fstruct%2Fnrf52832_ps.html>



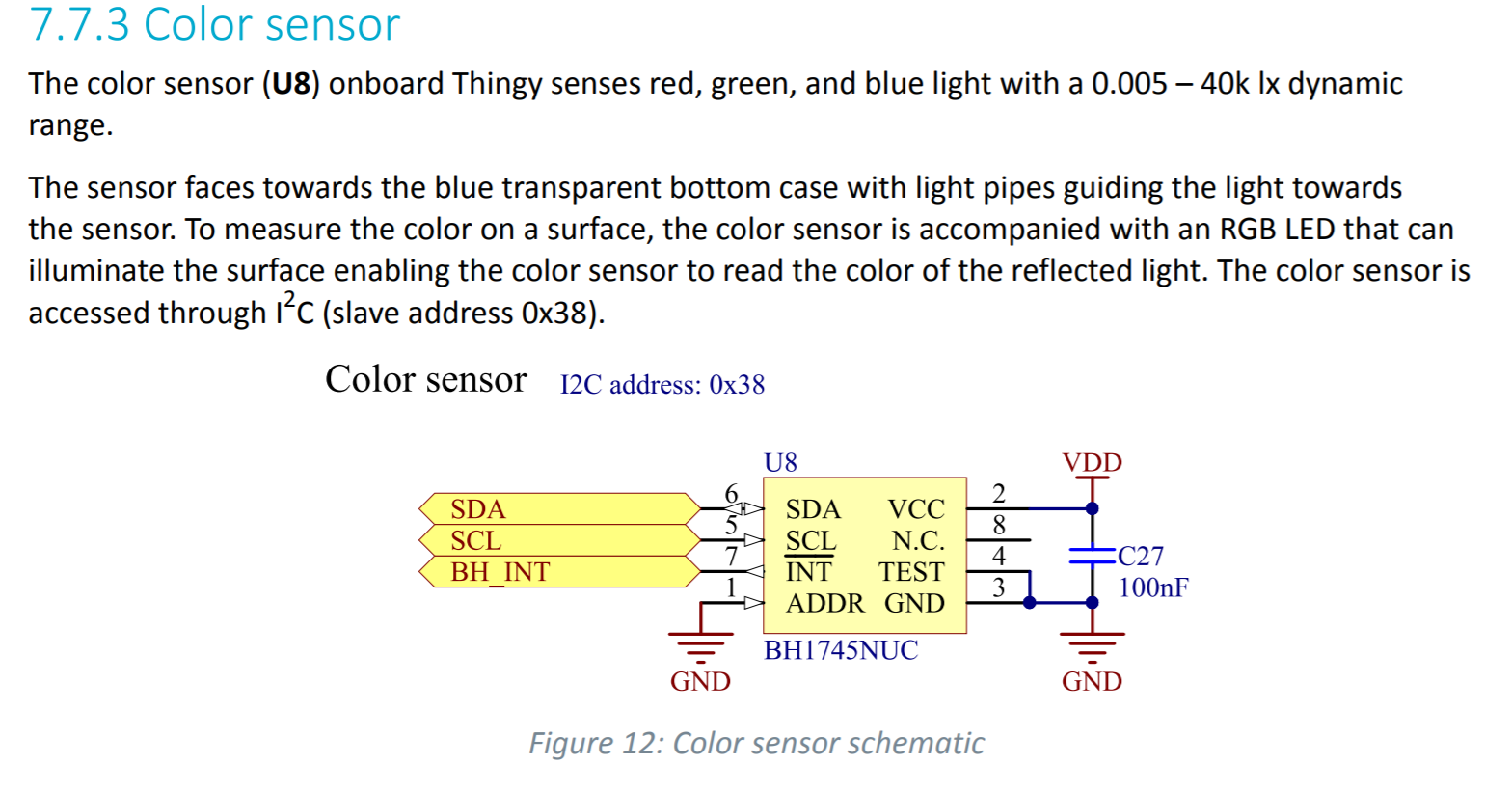


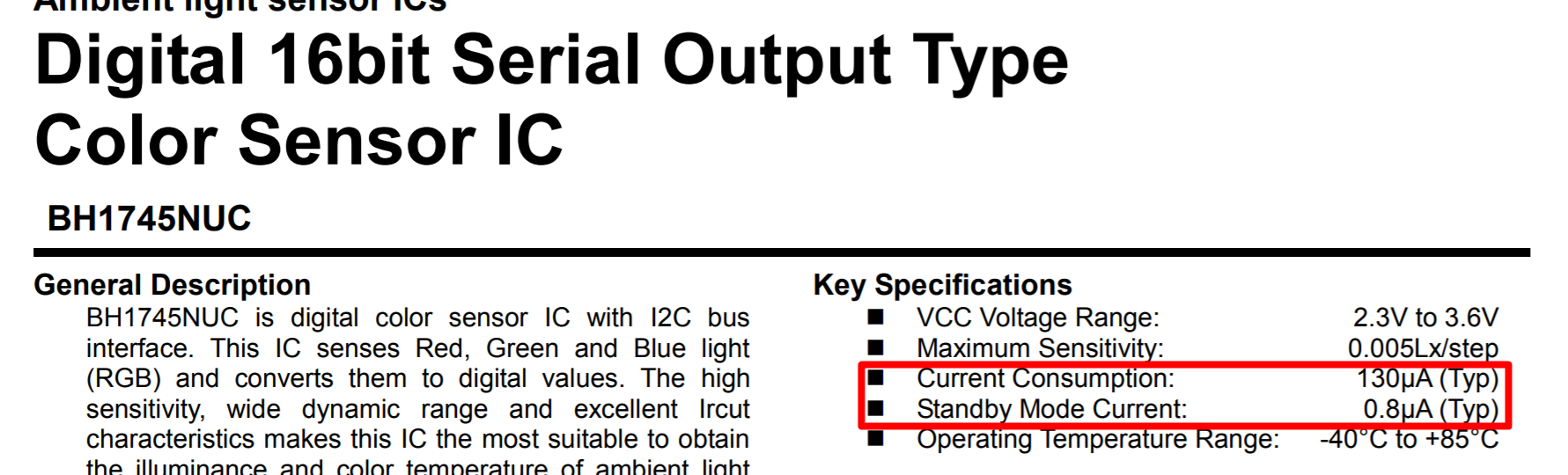
For the numbers below, I took the worst numbers possible.

* It is actually better to use the online energy profiler: <https://devzone.nordicsemi.com/nordic/power/w/opp/2/online-power-profiler-for-ble>

Light sensor

The thingy has a built in color sensor but it can sense the intensity as well. <https://www.mouser.co.uk/datasheet/2/348/bh1745nuc-e-519994.pdf>





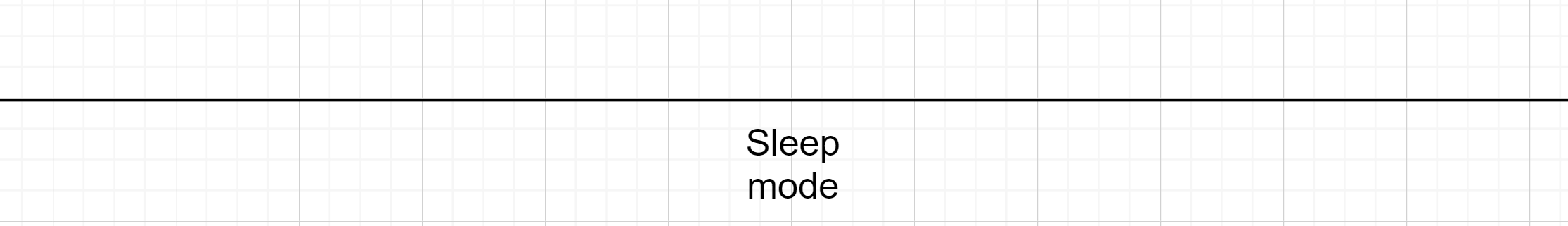
In the datasheet we can see that the normal current consumption is 130 µA. (🡺 130 µAh if one hour on)  
Standby mode current is 0,8µA (🡺 0,8µAh)

Total power consumption for a specific timeframe

This depends on the scheme we use. In the beginning, it wasn’t very clear what parameters we should take but now it is. We will scan 45 seconds because the octa can only send every 1m30 seconds. If we scan every 45 seconds, we will definitely scan 2 times in 1 interval of 1m30. This is because if we would fail to receive a packet then it has still another scan. So the accuracy is reasonable. The ofcourse depends on the error rate, accuracy needed, power consumption.

If we look at the power consumption scheme for the thingy’s we have at the moment there are 2 modes/schemes that will be used the most. So for these modes, the consumption will be estimated. These are sleep mode and on mode.

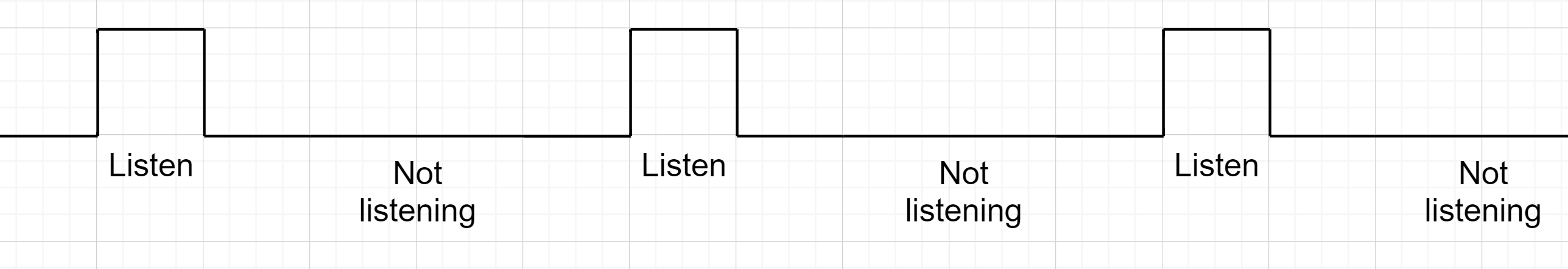
We will first discuss sleep mode:



*Fig 1: sleep mode*

When the thingy is in sleep mode => so nothing will consume power except the accelerometer and the nrf52832 but both are in low power mode. This mode will be on the longest (because most of the time the chair is not occupied e.g. during nighttime) which is preferable. The estimated power consumed here will be: 0,011 mA + 0,0019 mA = 0,0129mA ( = normal mode accelerometer + normal nrf52832) 🡺 for 1 hour= 0,0129mAh 🡺 for 12 h : 12 \* 0,0079mAh = 0,1548 mAh for 12 hours sleeping

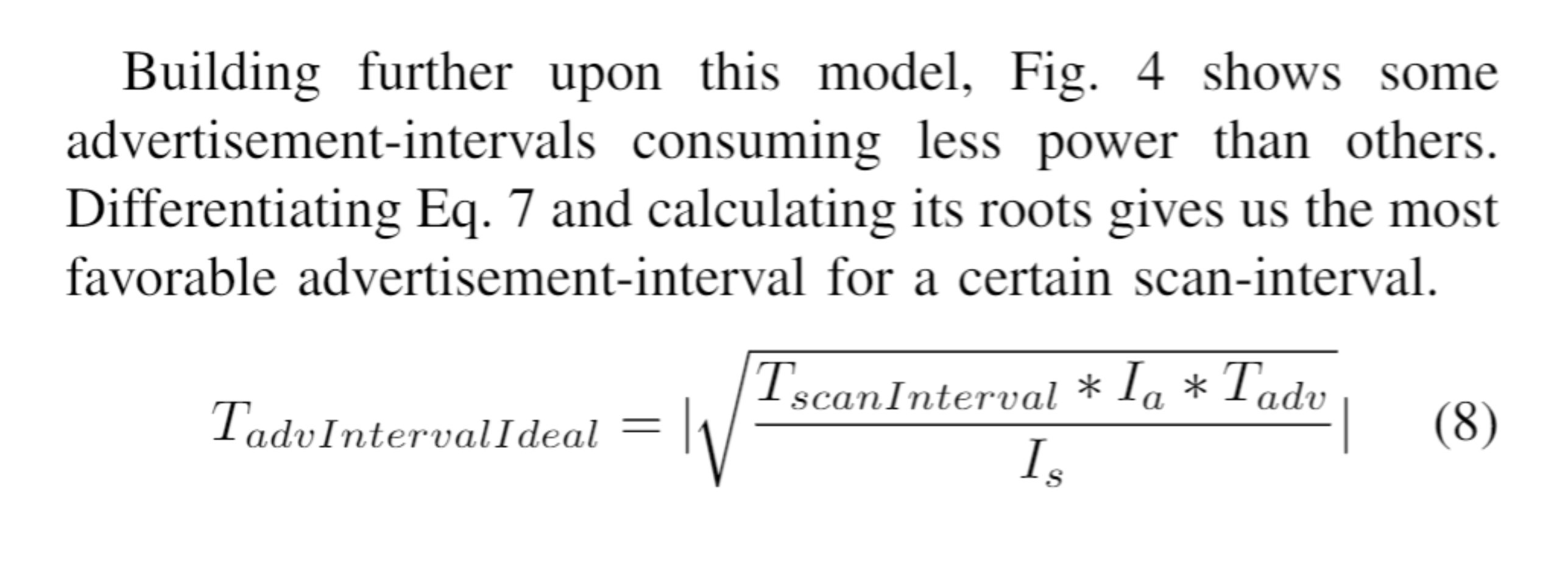
ON mode (= when someone is on the chair):



*Fig 2: On mode*

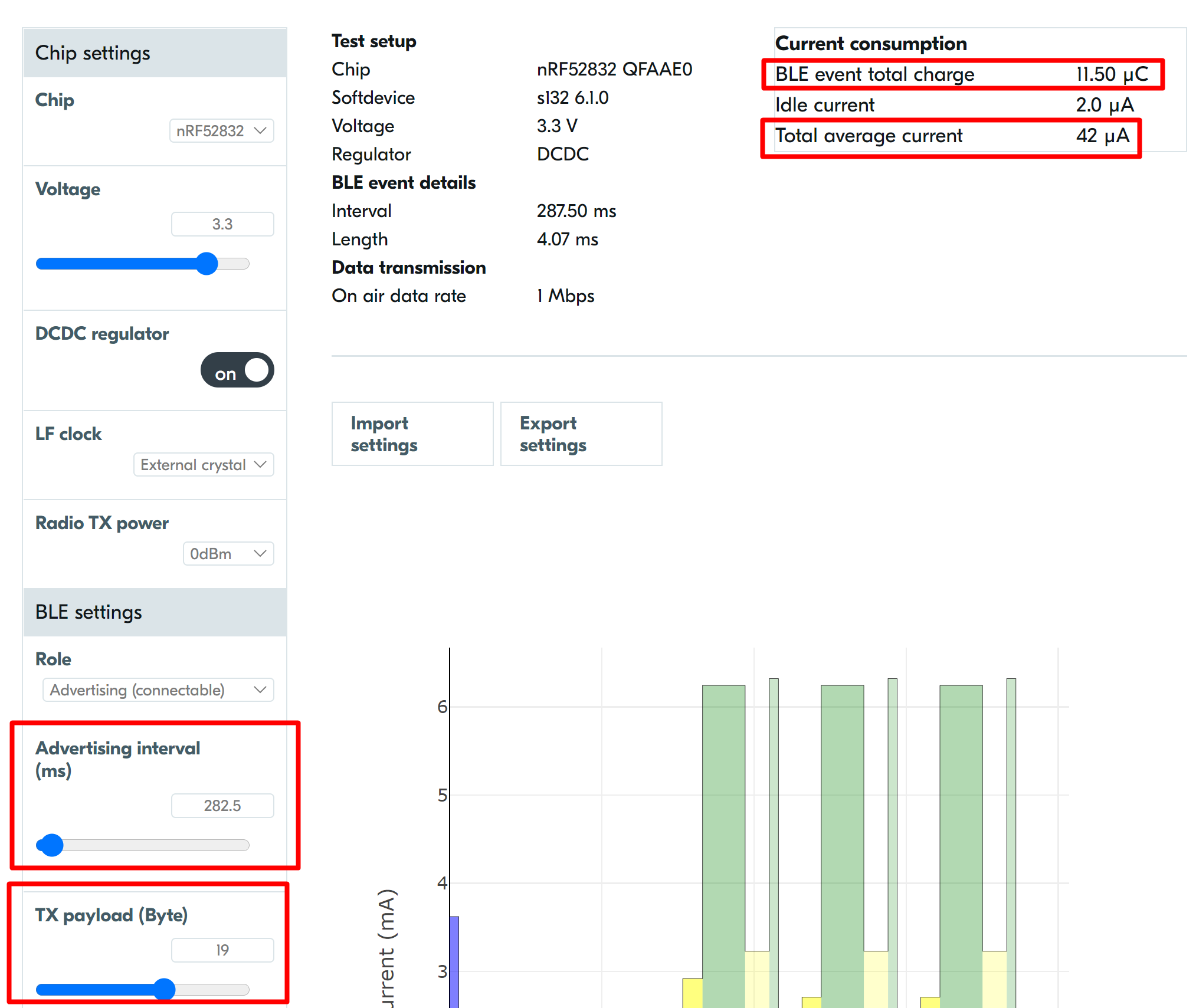
This mode consists of two parts: a long part where the light sensor just checks if someone leaves the chair and also advertises itself and a small part where the thingy listens for a few seconds to the other thingys to check who is closest and with what RSSi.

Calculations for the on mode

We want the scanwindow as small as possible => this means we need the advertisements as fast as possible. But this in turn raises also the power consumption => so what is the optimal advertisement interval (=scan window). In the paper from Ivan you can find in section IV power consumption (please read this) a similar situation where a balance need to be found how big the advertisement interval should be.   
The formula that we will use is this one:  


TscanInterval = 45 seconds(we look at these two options)  
Is = Scancurrent = 6.5 mA (see datasheet of nrf52832 a few pages back)  
Ia \* Tadv = BLE event total charge (you can find this easy in the BLE tool 🡺 this isn’t affected by advertisement interval, but it is affected by payload size => we use 19 bytes) = 11,50 µC

That means that:  
for 30 seconds TadvIntervalIdeal = sqrt(30\*0,00001150/0,0065)= 2821619829 ms  
In the BLE tool we se the following



* This means we get a power consumption of 42 µA when advertising

For our payload we use flags that I couldn’t get away, complete local name “Thingy” (this is for the octa so that it can filter on packets and doesn’t need all the packets coming in), manufacturing data (Type + SelfID + NeighbourID + RSSI).

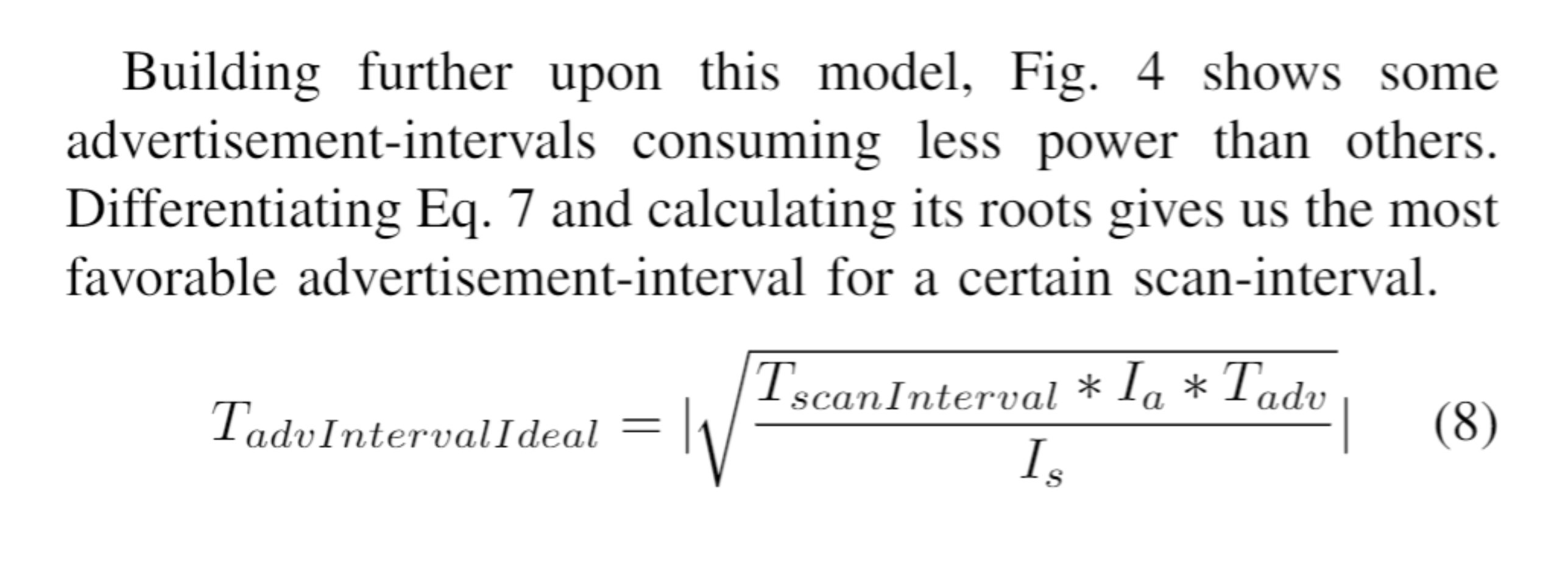
Not listining part (advertising): (current from nrf52832 + accelerometer + Light sensor)  
= 42 µA + 11µA + 130µA = 0,183 mA  
Listening part = 6.5 mA (see datasheet of nrf52832 a few pages back)

So how much does the on mode consume when somebody is sitting for an hour? 🡺 45 seconds are in not listening (advertising) mode and 0,2825 s in listening (scanning) mode => we have set the listenening time to 1 second so it can receive up to 3 packets in one scan window because there could be lots of thingy’s sending and so we hope to get the packet of the closest one. That means that we can calculate the power consumption for the separate parts in the timeframe and then convert it to an hour. After that we can add those two together to get the total power consumption per hour.   
🡺# timeframes in 1 hour = 3600 / (44+1) = 80 timeframes  
🡺 power consumption listening (scanning) per hour = 6,5mA \* (80 \* 1) / 3600 = 0,1444 mAh  
🡺 power consumption not listening (advertising) per hour = 0,183 mA \* (80 \* 44) / 3600 = 0,17893 mAh   
🡺total power consumption in an hour in on mode = 0,1444 mAh + 0,17893 mAh = 0,323377 mAh

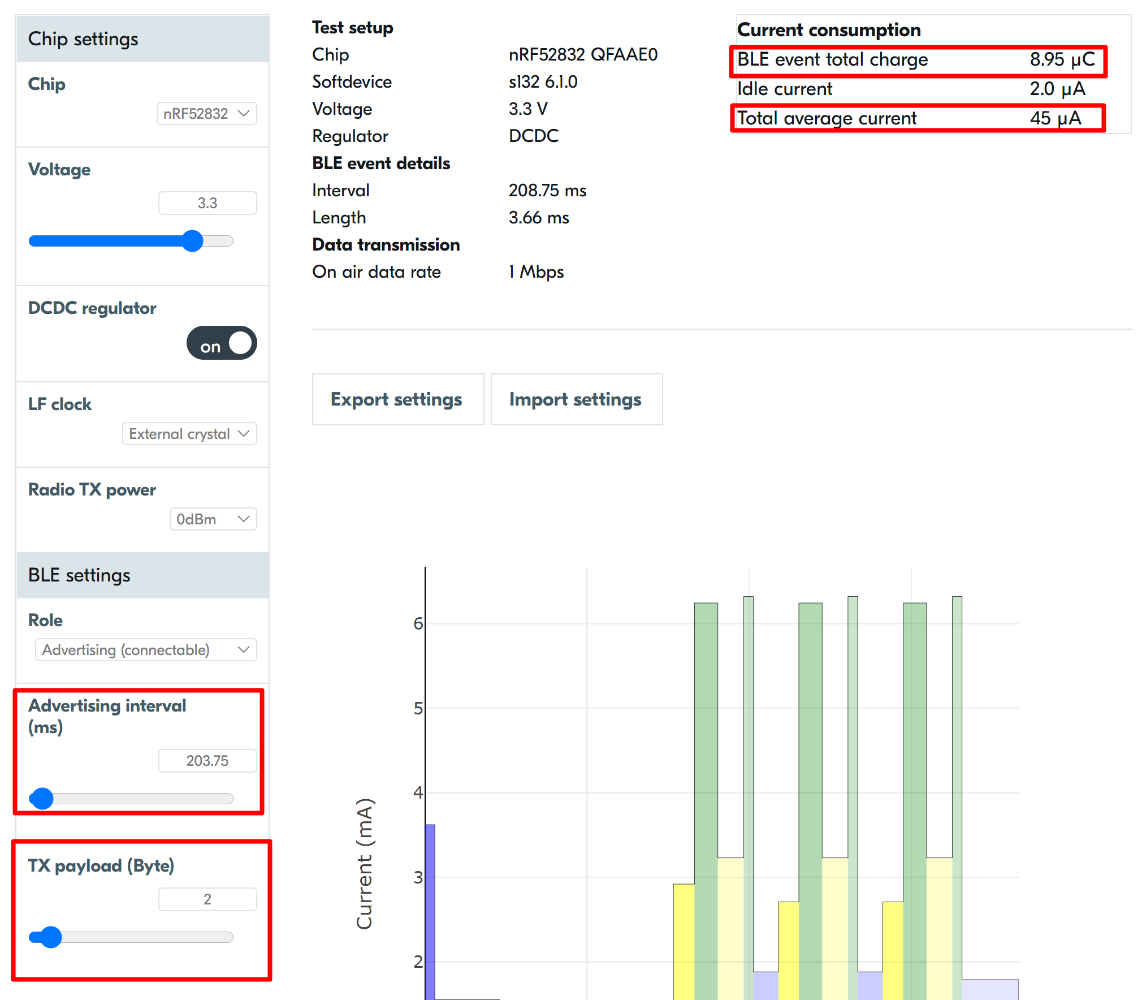
So for 12 hours in on mode 🡺 12 \* 0,323377 mAh = 3,8805 mAh

Battery has a capacity of 1440 mAh 🡺 that means if 12 hours of the day is in sleep mode and the other 12 is in on mode and for one day we have 3,8805 mAh + 0,1548 mAh = 4,0353 mAh for one 24 hour cycle 🡺1440 / 4,0343 = 356,85 days 🡺 lets assume 20 percent is lost by leakage and stuff 🡺 356,85 \* 0,8 = 285,48 days.

**This was our first estimation but it’s not used but I left it here:**

In the paper from Ivan you can find in section IV power consumption (please read this) a similar situation where a balance need to be found how big the advertisement interval should be.   
The formula that we will use is this one:  


TscanInterval = 30 seconds(we look at these two options)  
Is = Scancurrent = 6.5 mA (see datasheet of nrf52832 a few pages back)  
Ia \* Tadv = BLE event total charge (you can find this easy in the BLE tool 🡺 this isn’t affected by advertisement interval, but it is affected by payload size) = 8,95 µC

That means that:  
for 30 seconds TadvIntervalIdeal = sqrt(30\*0,00000895/0,0065)=203,2429 ms  
In the BLE tool we se the following

With the BLE tool, we can find the power consumption in not listening mode 🡺 = 45 µA

Where comes the Tx payload come from? 🡺 1 bye is for ID and 1 byte for flags ( e.g. first bit is 1 means a chair is violating the distance rule).

Not listining part (advertising): (current from nrf52832 + accelerometer + Light sensor)  
= 45 µA + 11µA + 130µA = 0,186 mA  
Listening part = 6.5 mA (see datasheet of nrf52832 a few pages back)

So how much does the on mode consume when somebody is sitting for an hour? 🡺 30 seconds are in not listening (advertising) mode and 0,205 seconds (0,2032429 rounded up to have a small margin) in listening (scanning) mode. That means that we can calculate the power consumption for the separate parts in the timeframe and then convert it to an hour. After that we can add those two together to get the total power consumption per hour.   
🡺# timeframes in 1 hour = 3600 / (30 + 0,205) = 119,18556530375 timeframes  
🡺 power consumption listening (scanning) per hour = 6,5mA \* (119,18556530375 \* 0,205) / 3600 = 0,0441152127131 mAh  
🡺 power consumption not listening (advertising) per hour = 0,186 mA \* (119,18556530375 \* 30) / 3600 = 0,1847376262208 mAh   
🡺total power consumption in an hour in on mode = 0,0441152127131 mAh + 0,1847376262208 mAh = 0,2288528389339 mAh

So for 12 hours in on mode 🡺 12 \* 0,2288528389339 mAh = 2,74623 mAh

Battery has a capacity of 1440 mAh 🡺 that means if 12 hours of the day is in sleep mode and the other is in on mode and for one day we have 2,74623 + 0,1548 mAh = 2,901 mAh for one 24 hour cycle 🡺1440 / 2,901 = 496,3737 days 🡺 lets assume 20 percent is lost by leakage and stuff 🡺 496,3737 \* 0,8 = 397,0997 days.