* **ESP32 Stockage :** 520 KB of on-chip SRAM for data and instructions.
* **8 KB** of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
* **8 KB** of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.

**Temperature values :**

1 feedback= 1 float = 4B

**Accelerometer/Gyroscope values :**

1 feedback = 6 floats = 24B

* code size : 206KB (memoire flash)

data storage and batch sending: aquisition frequency 4Hz

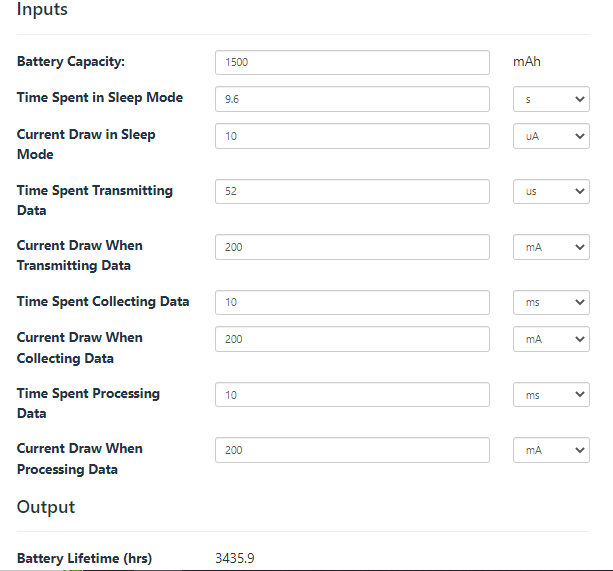
**chaque 10s :**

* temps de lecture : 0.4 s
* sleep mode 9.6

10s = 40 valeurs = 1KB

débit transmission de données via wifi : 19MB/s

à transmettre : 1KB

= 143 jours

**Data cleaning au niveau du collier :**

ESP32 is equipped with a dual-core processor that can run up to 240 MHz, which makes it capable of handling complex data processing tasks.

The suitability of ESP32 for data cleaning will depend on the specific requirements of the use case. Our data cleaning needs are rather simple , so the ESP32 can handle the tasks.

The data cleaning process takes time and consumes energy which reduces the collar’s battery life.

It's worth noting that the actual data cleaning process would typically occur on the server-side rather than on the collar itself, as the server typically has more processing power and storage capacity. The role of the collar would be to collect and transmit the raw data to the server for processing and analysis.

\_\_\_\_\_\_\_\_\_\_\_\_

Battery Life = Battery Capacity / [(Current Consumption of Controller + Current Consumption of Temperature Sensor + Current Consumption of Accelerometer) x Duty Cycle + Self-Discharge Current]

Duty Cycle = The percentage of time the device spends in active mode during a given period

Self-Discharge Current = The amount of current consumed by the battery during sleep mode. It can be estimated by considering the specifications of the battery and the quiescent current of the ESP32 board during sleep mode.

**Example**

To calculate the consumption of the controller, we can use the following formula:

Controller Consumption = 300 mA (during active mode) x Duty Cycle + Quiescent Current (during sleep mode)

Assuming a quiescent current of 20 uA during sleep mode, and a duty cycle of 1.11%, the controller consumption can be calculated as follows:

Controller Consumption = 300 mA x 1.11% + 20 uA

Controller Consumption = 3.33 mA + 20 uA

Controller Consumption = 3.35 mA

To calculate the current consumption of the temperature sensor, we can use the formula:

Temperature Sensor Consumption = 1 mA x Duty Cycle

Assuming a duty cycle of 1.11%, the current consumption of the temperature sensor can be calculated as follows:

Temperature Sensor Consumption = 1 mA x 1.11%

Temperature Sensor Consumption = 0.0111 mA

To calculate the current consumption of the accelerometer, we can use the formula:

Accelerometer Consumption = 200 uA x Duty Cycle

Assuming a duty cycle of 1.11%, the current consumption of the accelerometer can be calculated as follows:

Accelerometer Consumption = 200 uA x 1.11%

Accelerometer Consumption = 2.22 uA

The self-discharge current can be estimated based on the battery specifications and the quiescent current of the controller during sleep mode. Assuming a quiescent current of 20 uA during sleep mode, and a battery capacity of 1600 mAh, the self-discharge current can be calculated as follows:

Self-Discharge Current = Battery Capacity / (365 x 24 x 3600) x Quiescent Current

Self-Discharge Current = 1600 mAh / (365 x 24 x 3600) x 20 uA

Self-Discharge Current = 0.00000000221 A or 2.21 nA

Finally, we can calculate the duty cycle using the following formula:

Duty Cycle = Active Time / (Active Time + Sleep Time)

Assuming the system is active for 10 seconds every 15 minutes, the duty cycle can be calculated as follows:

Active Time = 10 seconds

Sleep Time = 15 minutes - 10 seconds = 899 seconds

Duty Cycle = 10 / (10 + 899) x 100

Duty Cycle = 1.11%

Using the values calculated above, we can estimate the expected battery life as follows:

Battery Life = Battery Capacity / [(Controller Consumption + Temperature Sensor Consumption + Accelerometer Consumption) x Duty Cycle + Self-Discharge Current]

Battery Life = 1600 mAh / [(3.35 mA + 0.0111 mA + 2.22 uA) x 1.11% + 2.21 nA] = 109.72 days