

STM32CubeMX

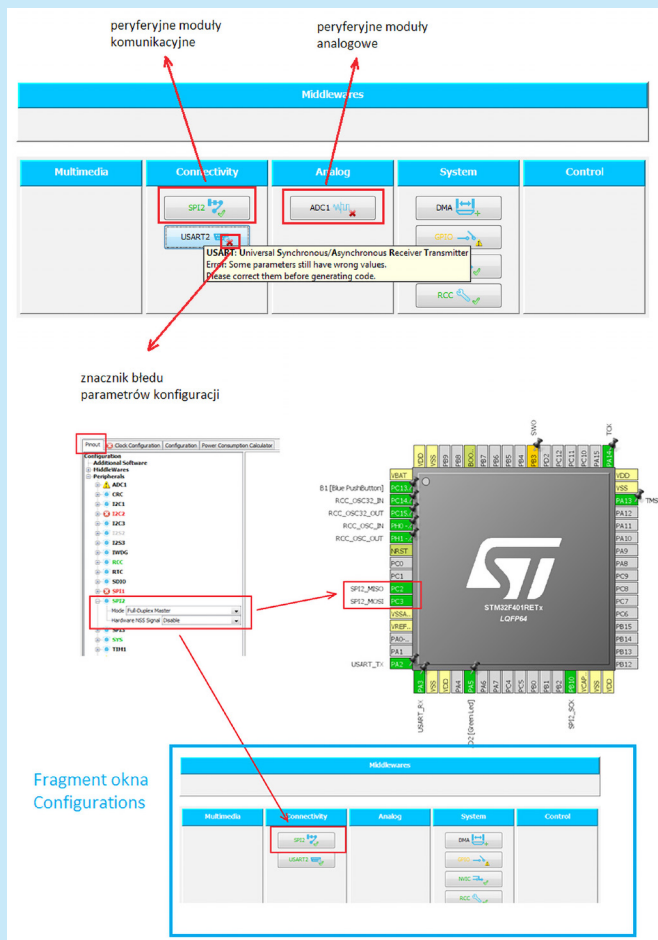
– graficzny konfigurator STM32 krok-po-kroku (2)

Prezentowany w ostatnich numerach EP pakiet narzędziowy Atollic TrueSTUDIO for STM32 zapewnia kompletne wsparcie w zakresie edycji, kompilacji i debugowania kodu. Teraz zajmujemy się przybliżeniem możliwości pakietu STM32Cube, który jest bezpłatnym konfiguratory mikrokontrolerów STM32.

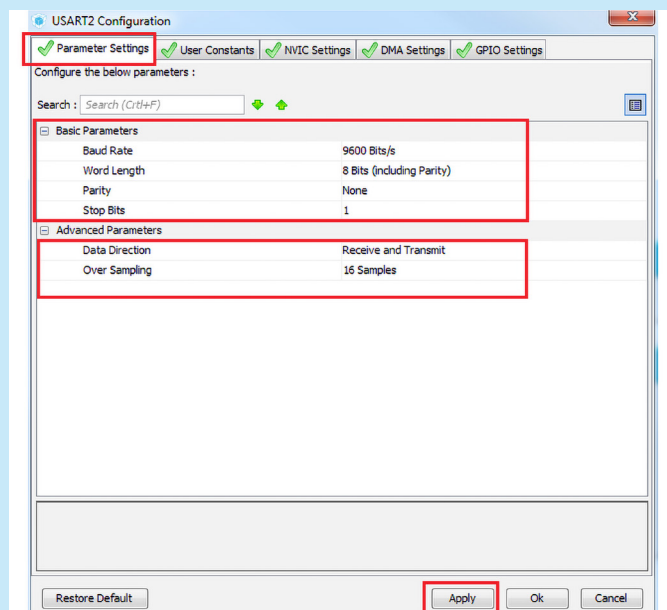
Po tym jak w zakładce Pinouts wybierzemy moduł peryferyjny i zostanie dla niego przydzielone wyprowadzenie można będzie skonfigurować szczegółowe parametry pracy tego modułu w zakładce Configurations (**rysunek 23**). Moduły peryferyjne, które na liście okna Pinout są oznaczone na zielono są automatycznie umieszczane w odpowiednich dla nich polach zakładki Configuration. Przy każdym symbolu modułu umieszczono znacznik określający stan konfiguracji. Na rys. 23 wskaźnik błędu wygenerowany celowo do celów testowych jest umieszczony przy module UART i ADC1.

Kliknięcie na ikonę modułu peryferyjnego w zakładce Configuration otwiera okno z parametrami do konfiguracji modułu. Na rysunku 24 pokazano okno dla modułu USART2 zawierające zakładki:

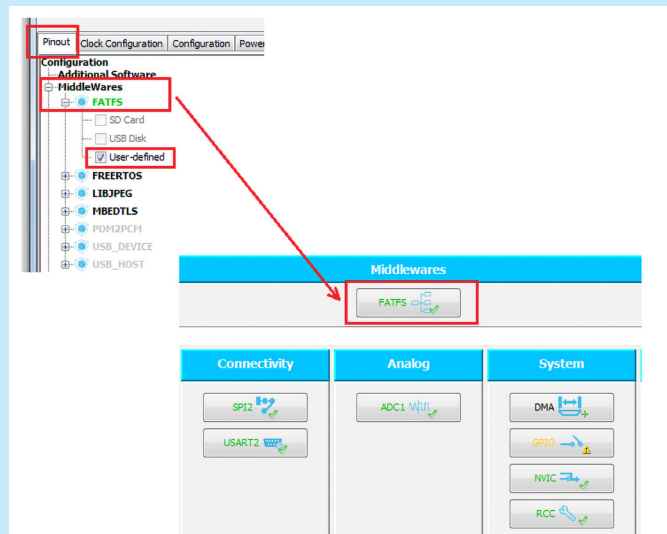
- **Parameter settings** – przeznaczona do ustawiania parametrów transmisji: prędkość, liczbę bitów/ramkę, parzystość, oraz kierunku przesyłania danych.



Rysunek 23. Zakładka okna Configuration

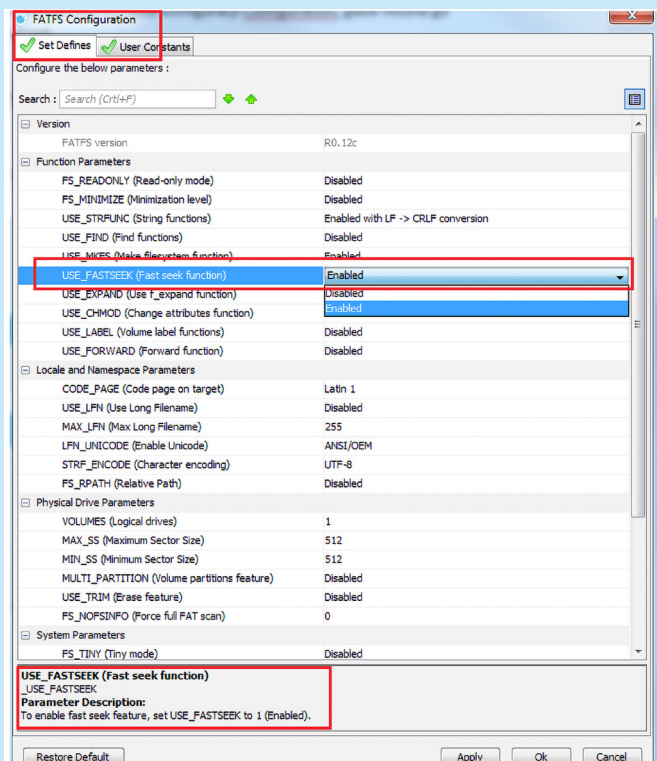


Rysunek 24. Okno konfiguracji modułu USART2 z zakładką Parameter Settings



Rysunek 25. Dodanie modułu programowego FATFS

- **User Constans** – przeznaczona do definiowania stałych użytkownika.
- **NVIC Settings** – przeznaczona do konfigurowania przerw powiązanych z pracą modułu USART2.



Rysunek 26. Konfigurowanie modułu FATFS

- **DMA Settings** przeznaczona do konfigurowania kanału DMA (jeżeli ma być użyty).
- **GPIO Settings** – przeznaczona do uzupełniającego konfigurowania linii I/O modułu USART2.

W taki sposób można skonfigurować dla własnych potrzeb wszystkie używane moduły konfiguracyjne.

Wróćmy na chwilę do zakładki Pinouts. Umieszczono tam oprócz listy Peripherals również listę MiddleWares z możliwością dodawania do projektu modułów programowych warstwy wyższej. Do celów testowych dodamy obsługę systemu plików FATFS (**rysunek 25**). Podobnie jak w przypadku układów peryferyjnych, moduł programowy FATFS jest dodawany do okna konfiguracji Configuration, w którym można go skonfigurować zależnie od potrzeb.

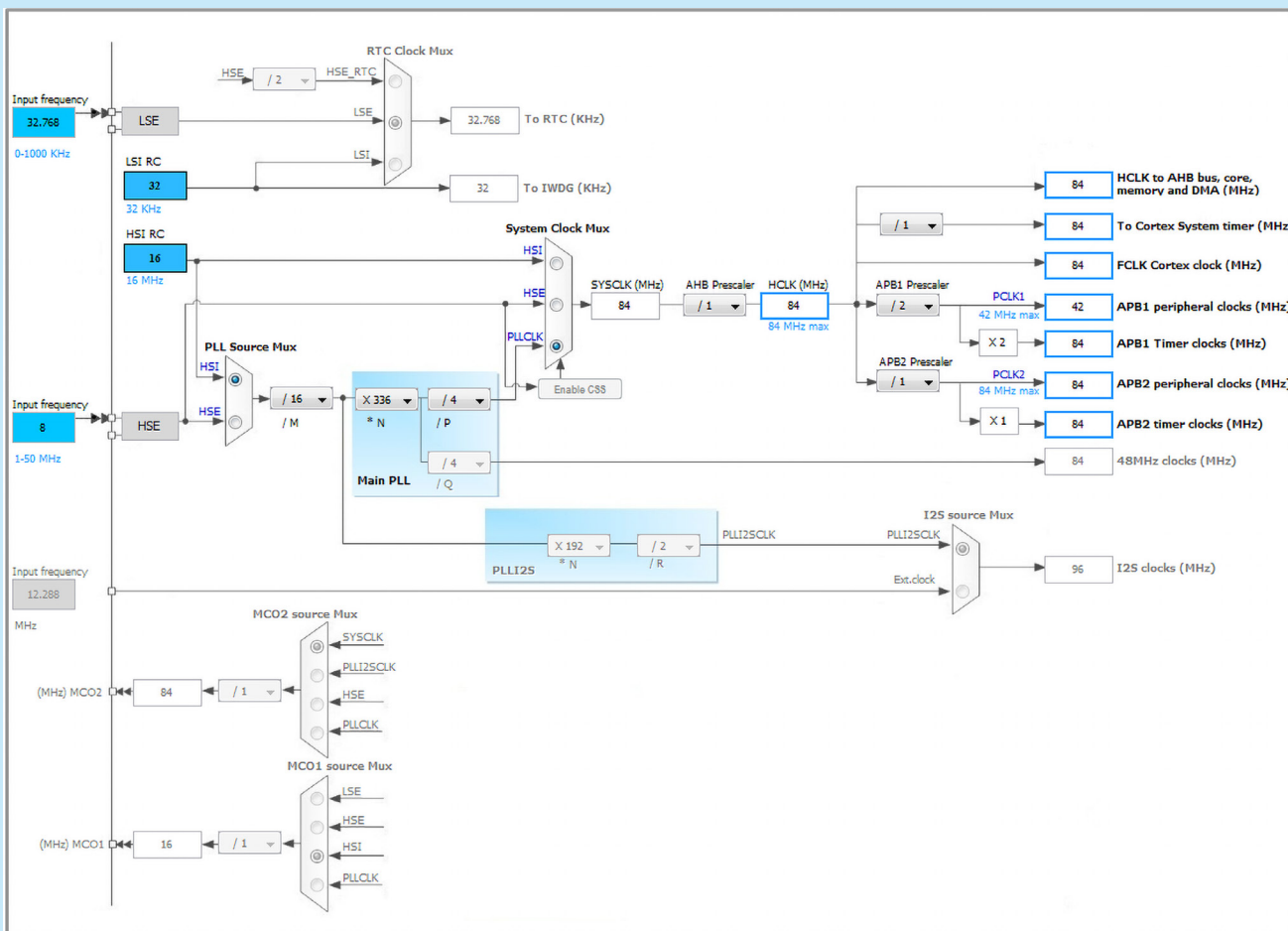
Okno konfiguracji FATFS otwierane po kliknięciu na ikonę FATFS umieszczoną w oknie konfiguracji (Middlewares) pokazano na **rysunku 26**.

Konfigurowanie układu taktowania

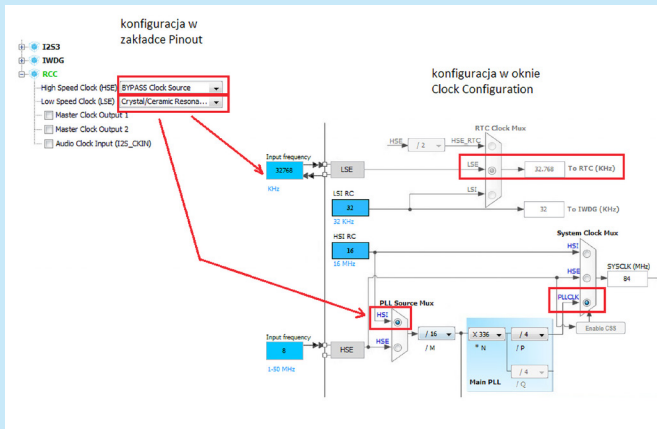
Do konfigurowania układu taktowania jest przeznaczony okno Clock Configuration pokazany na **rysunku 27**. Ustawianie parametrów taktowania jest powiązane z modułem konfiguracyjnym RCC umieszczonym w liście Peripherals okna Pinouts (**rysunek 28**). Źródłem zegara taktującego układami mikrokontrolera może być:

- Wewnętrzny, precyzyjny oscylator RC o częstotliwości 16 MHz.
- Oscylator stabilizowany zewnętrznym rezonatorem kwarcowym o częstotliwości z zakresu od 1 do 50 MHz (w praktyce najczęściej jest 8 MHz).

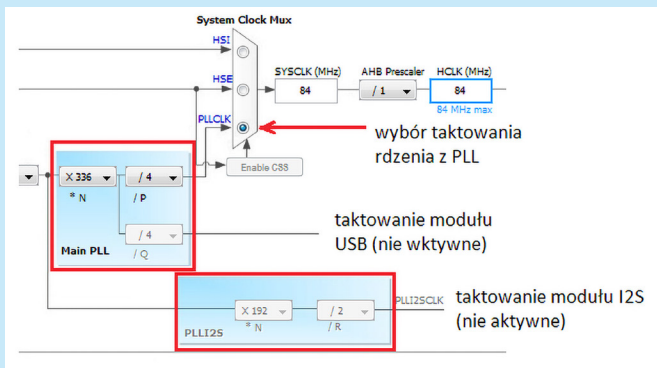
Rdzeń mikrokontrolera z rodziny STM32F4 zastosowanego w naszym projekcie może być taktowany z maksymalną częstotliwością



Rysunek 27. Okno konfiguracji taktowania



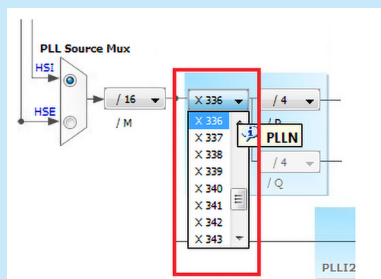
Rysunek 28. Powiązanie ustawień okna Pinouts i okna Clock Configuration



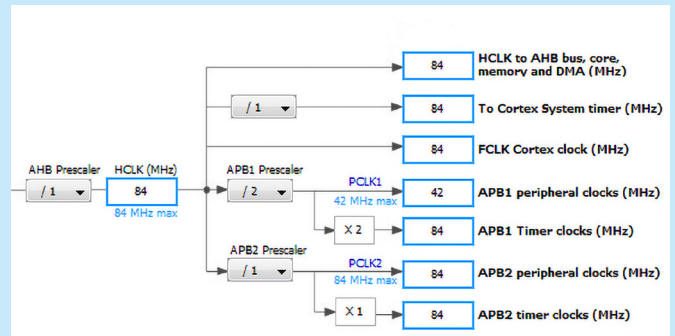
Rysunek 29. Układy PLL

84 MHz. Taką wartość można uzyskać z częstotliwości oscylatorów wbudowanych w mikrokontroler (wewnętrznego RC lub zewnętrznego stabilizowanego kwarcem) przez powielanie ich częstotliwości za pomocą PLL. Wbudowany w układ taktowania moduł PLL umożliwia ustawianie częstotliwości w szerokim zakresie. Sygnał z jego wyjścia może być źródłem taktowania rdzenia i magistral układów peryferyjnych, taktowania układu USB (jeżeli jest zabudowany) i modułu komunikacyjnego I²S przeznaczanego do przesyłania danych audio.

Ustawianie częstotliwości odbywa się z poziomu okna przez kliknięcie na ikony mnożnika *N o dzielnika /P (rysunek 30). Graficzny interfejs programujący taktowanie jest bardzo intuicyjny. Pozwala na proste i szybkie dobranie częstotliwości taktowania do swoich potrzeb. Bardzo pomocne jest graficzne przedstawienie ustawionych częstotliwości taktowania: rdzenia, magistrali AHB i układów peryferyjnych. Pokazano to na rysunku 31. Ustawienia są wyświetlane online. Kiedy konfigurator wykryje błędy w konfiguracji taktowania, to są one sygnalizowane na czerwono, jak pokazano na rysunku 32. Błędy te mogą wynikać z przekroczenia maksymalnej częstotliwości dopuszczalnej dla danego typu rdzenia czy magistrali peryferii.



Rysunek 30. Programowanie układu PLL taktującego rdzeń i peryferia



Rysunek 31. Graficzne przedstawienie częstotliwości taktowania rdzenia i magistrali APBx

Oprócz taktowania rdzenia i peryferii można wybierać źródło sygnału taktującego małej częstotliwości dla układu zegara RTC. Może to być zewnętrzny oscylator stabilizowany kwarcem zegarkowym 32,768 kHz, lub wbudowany oscylator RC o częstotliwości 32 kHz.

Symulacja poboru mocy

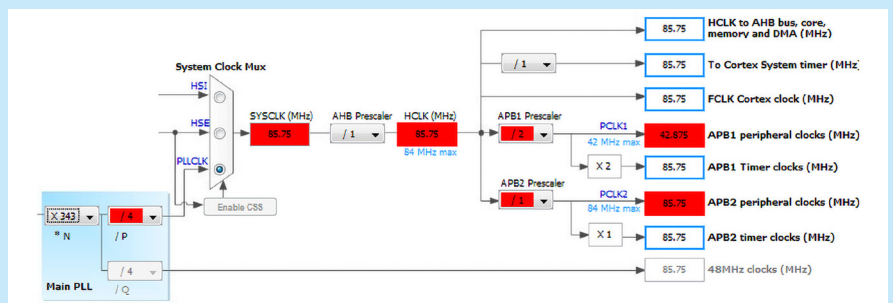
W układach z mikrokontrolerami coraz częściej pojawia się wymaganie redukcji pobieranej mocy ze źródła zasilania. Jest to naturalne w układach zasilanych z baterii. STM32CubeMX ma wbudowany kalkulator do obliczania poboru mocy umieszczony w zakładce Power Consumption Calculator.

Consumption Calculator umożliwia obliczenie:

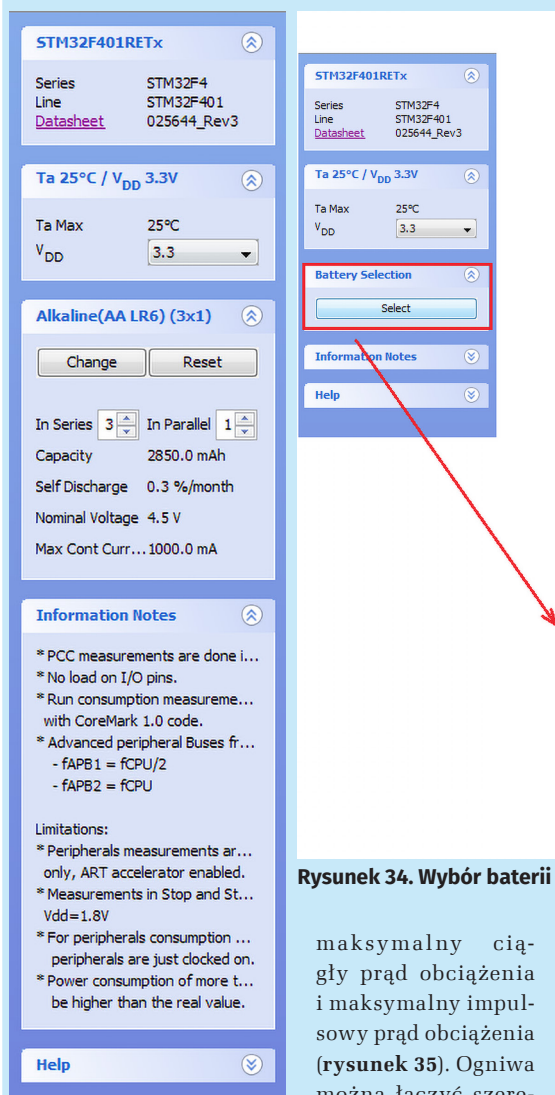
- Średniego zużycia prądu.
- Żywotności baterii.
- Wydajności mikrokontrolera wyrażonej w DMPIS.
- Maksymalnej temperatury otoczenia zapewniającej dobre warunki pracy układu.

Okno kalkulatora oprócz wizualizacji szacunkowego zużycia energii, umożliwia też projektantowi wybieranie metod jej obniżania poprzez stosowanie trybów obniżonego poboru mocy (jeżeli są dostępne), doboru minimalnej możliwej częstotliwości taktowania do wykonania zadania i włączanie zasilania tylko tych układów peryferyjnych, które są w danej chwili niezbędne. Każda z tych metod może być oddzielnie konfigurowana zależnie od potrzeb. Symulacja może uwzględniać zasilanie układu z magistrali USB (VBUS) lub z baterii. W wypadku użycia baterii można wykonać symulację z różnymi napięciami baterii odpowiednimi do stanu naładowania.

Wylizanie poboru mocy odbywa się w krokach (step) definiowanych przez użytkownika. W oknie zakładki po lewej stronie jest obszar zaznaczony na niebiesko, w którym można wybrać napięcie zasilania V_{dd} z zakresu 1,8...3,3 V oraz wybrać rodzaj baterii zasilającej (rysunek 33). Wybór napięcia zasilania nie wymaga komentarza. W razie zasilania z baterii mamy możliwość wybrania jednego z predefiniowanych ustawień lub zdefiniowania własnej „baterii” (rysunek 34). Podajemy przy tym nazwę definicji, pojemność w mAh, procent rozładowania/miesiąc, napięcie nominalne,



Rysunek 32. Sygnalizacja błędów przekroczenia maksymalnych częstotliwości taktowania



Rysunek 34. Wybór baterii zasilającej i opcjonalne definiowanie własnej baterii

Rysunek 33. Obszar wyboru napięcia zasilającego i zasilania bateryjnego

maksymalny ciągły prąd obciążenia i maksymalny impulsowy prąd obciążenia (rysunek 35). Ogniwa można łączyć szeregowo lub równoległo.

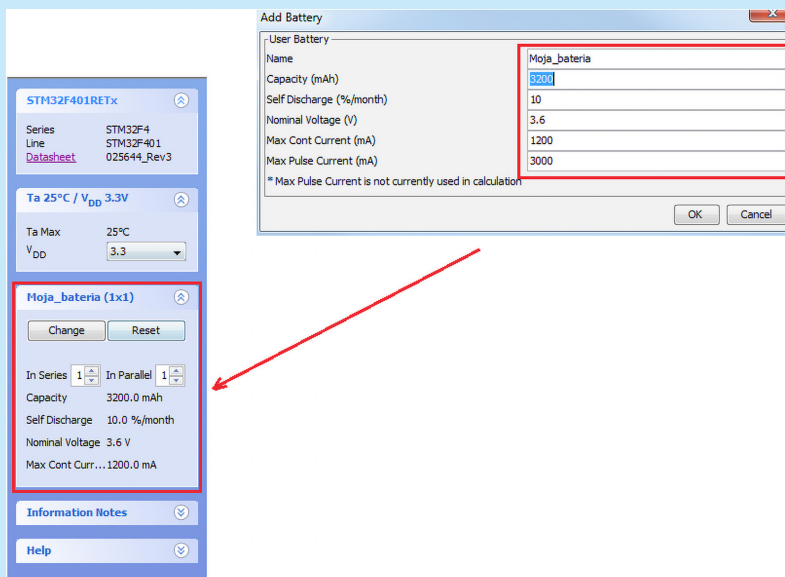
Na rysunku 36 pokazano definicję baterii zbudowanej z szeregowego połączenia 3 ogniw alkalicznych o napięciu +1,5 V.

Krok obliczeń poboru mocy dla konkretnej konfiguracji zasilania i konfiguracji pracy mikrokontrolera jest definiowany po kliknięciu na ikonkę zielonego plusa w oknie Step. Otwiera się wtedy okno New Step, w którym możemy wybrać w oknie Power/memory:

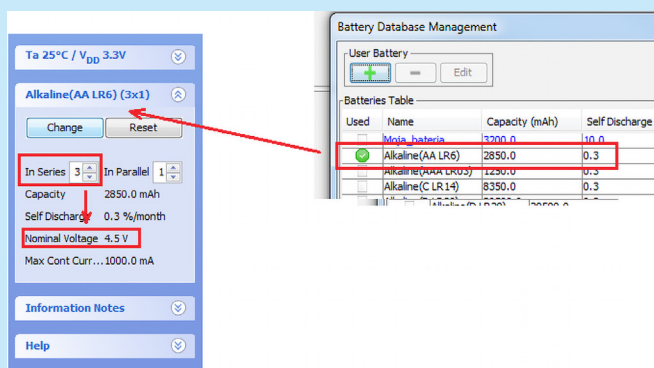
- **Power Mode** – czyli rodzaj pracy mikrokontrolera z uwzględnieniem trybów oszczędzania energii na przykład RUN, SLEEP, STANDBY, BATTERY.
- **Power Scale** – do wyboru są dwie opcje SCALE 2 Medium i SCALE3 – Low.
- **Memory FetchType** – sposób pobierania wykonywanego kodu: z pamięci SRAM, lub z pamięci FLASH.
- **VDD** – do wyboru jest napięcie +3,3 V lub +1,8 V.
- **Voltage Source** – możemy symulować zasilanie z baterii lub z magistrali Vbus.

Obliczenia są wykonywane na podstawie wybranych opcji. Warto przy tym pamiętać, że jest to symulacja i że w rzeczywistej aplikacji ostateczny pobór prądu zależy też od dołączonych do mikrokontrolera, współpracujących z nim komponentów – nie tylko aktywnych, ale też biernych.

Tomasz Jabłoński, EP



Rysunek 35. Definiowanie własnego ogniwa



Rysunek 36. bateria 3 ogniw typu AA (LR6)