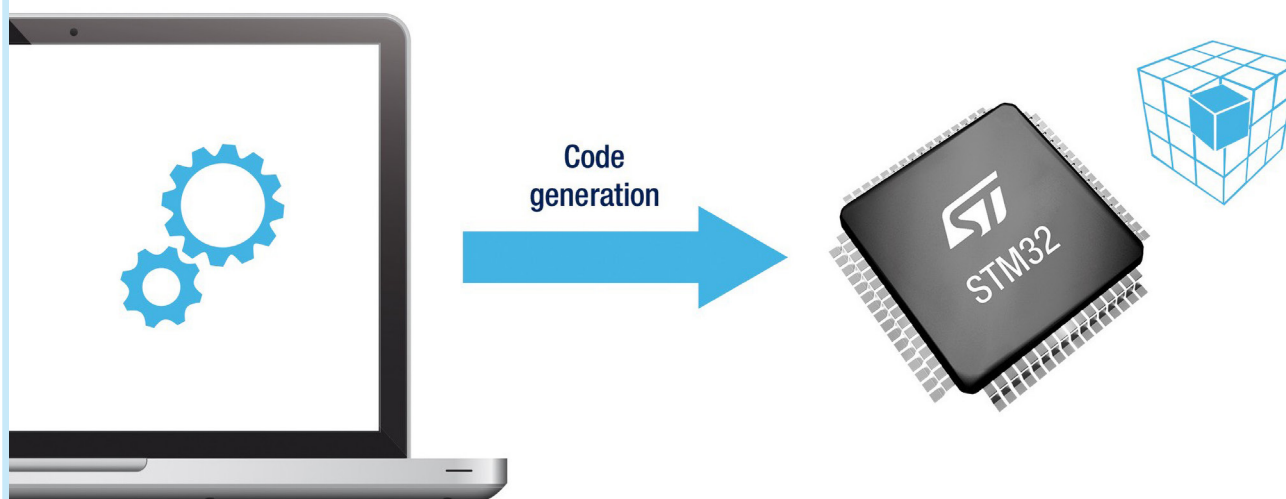


STM32Cube™ eases STM32 development



STM32CubeMX – graficzny konfigurator STM32 krok-po-kroku (3)

Prezentowany w ostatnich wydaniach „Elektroniki Praktycznej” pakiet narzędziowy Atollic TrueSTUDIO for STM32 zapewnia kompletne wsparcie w zakresie edycji, kompilacji i debugowania kodu. Teraz zajmiemy się przybliżeniem możliwości pakietu STM32Cube, który jest bezpłatnym konfiguratorem mikrokontrolerów STM32.

Jednym z ważnych zagadnień konfiguracji mikrokontrolerów STM32 jest ustalenie częstotliwości taktowania bloków peryferyjnych oraz CPU, które mają duże znaczenie dla wypadkowego poboru prądu. Ze względu na skomplikowaną budowę systemu taktującego, pakiet STM32CubeMX jest na tym etapie przygotowywania projektu bardzo przydatny.

Zaczynamy od okna Clocks, w którym widać następujące elementy (rysunek 37):

- **CPU Frequency** – można wybrać predefiniowane wartości lub wybrać opcję User Defined umożliwiającą dokładne ustawienie taktowania w zakresie od 30 do 60 MHz.
- **InterpolationRanges** – zakres ustawiania częstotliwości przez użytkownika.
- **ClockConfiguration i Clock Source Frequency** – konfiguracja źródła przebiegu taktującego.

Okno Optional Settings pozwala na podanie czasu, w którym krok jest wykonywany (Step Duration), oraz zasymulowanie dodatkowego poboru mocy na przykład przez zewnętrzne układy. Okno Peripherals pozwala oszacować pobór prądu przez każdy włączony układ peryferyjny.

Wynik symulacji poboru mocy jest wyświetlany w oknie Results:

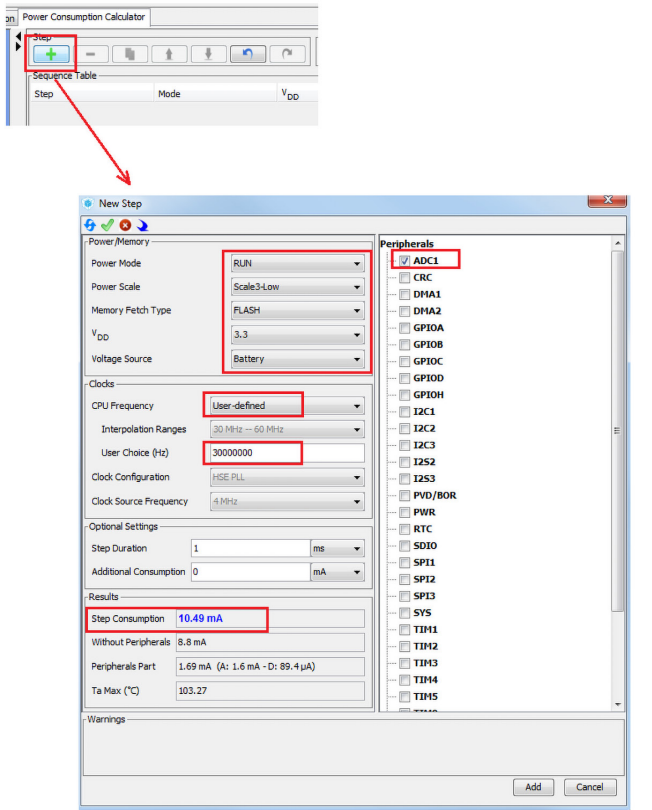
- **Step Consumptions** – całkowity pobór prądu dla ustalonej konfiguracji kroku.
- **WithoutPeripherals** – poboru prądu z wyłączonymi układami peryferyjnymi.
- **Peripheral Part** – pobór prądu przez układy peryferyjne.
- **TaMax** – maksymalna temperatura struktury układu, przy której mikrokontroler jeszcze pracuje poprawnie.

Po kliknięciu na przycisk Add nowy krok jest dodawany do listy kroków w oknie SequenceTable.

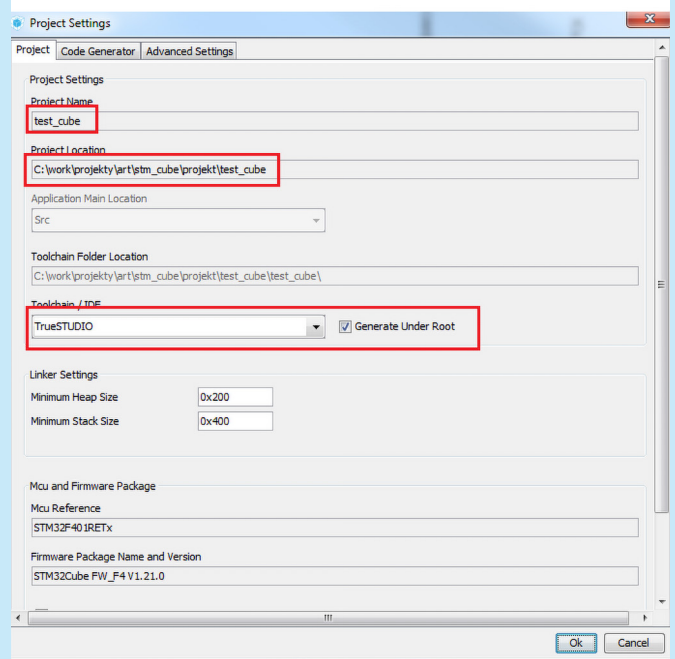
Na **rysunku 38** pokazano zdefiniowane dwa kroki: pierwszy z trybem pracy RUN i drugi z trybem pracy SLEEP. Krok 1 w trybie RUN trwa 20 sekund, a krok 2 w trybie SLEEP 10 sekund. W oknie Display jest wyświetlany wykres poboru prądu przez czas trwania obu zdefiniowanych kroków. Okno Display może opcjonalnie wyświetlać wykonywanie zdefiniowanych kroków w innej postaci – na przykład takiej jak została pokazana na **rysunku 39**.

Generowanie kodu wynikowego

Graficzny interfejs konfiguratora CubeMX jest zgodnie z założeniami przyjazny dla użytkownika, ale wynik konfiguracji musi być

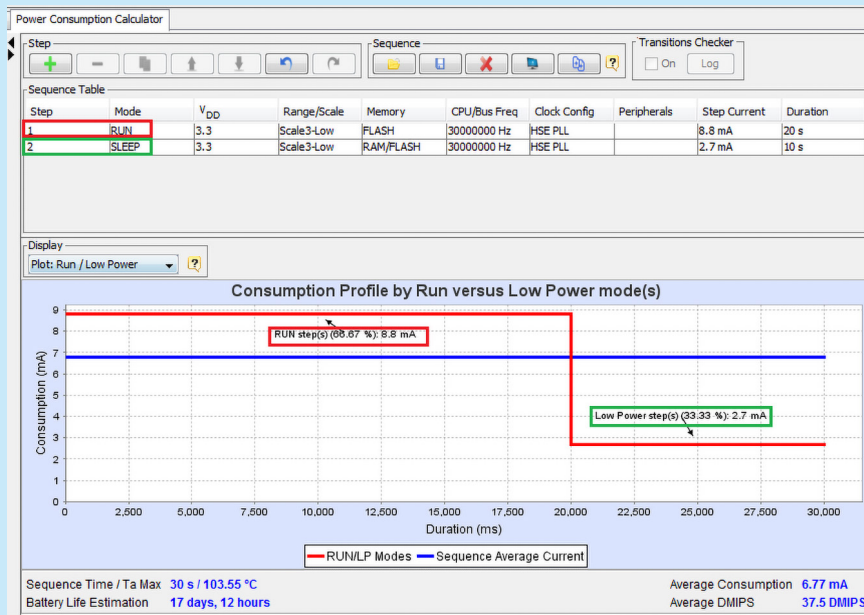


Rysunek 37. Konfiguracja kroku wyliczania zapotrzebowania na energię

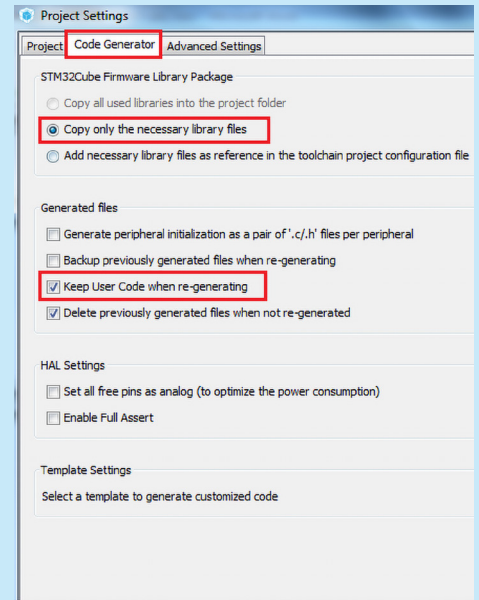


Rysunek 40. Zakładka Project okna Project Settings

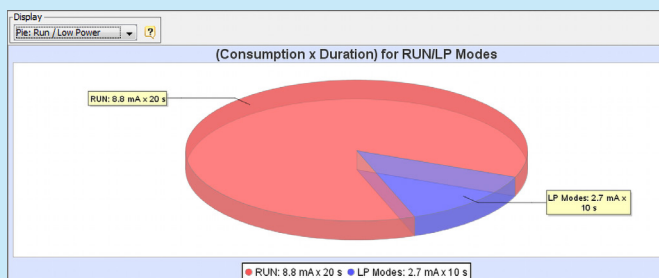
w końcowej fazie sprowadzony do postaci kodu źródłowego języka C. Generator kodu wykonuje kilka ważnych czynności. Po pierwsze sprawdza, czy zostały zainstalowane biblioteki dla wybranej rodziny mikrokontrolerów. Jeżeli nie, to biblioteka zostanie automatycznie pobrana i zainstalowana.



Rysunek 38. Przykładowa definicja dwu kroków w różnych trybach oszczędzania energii



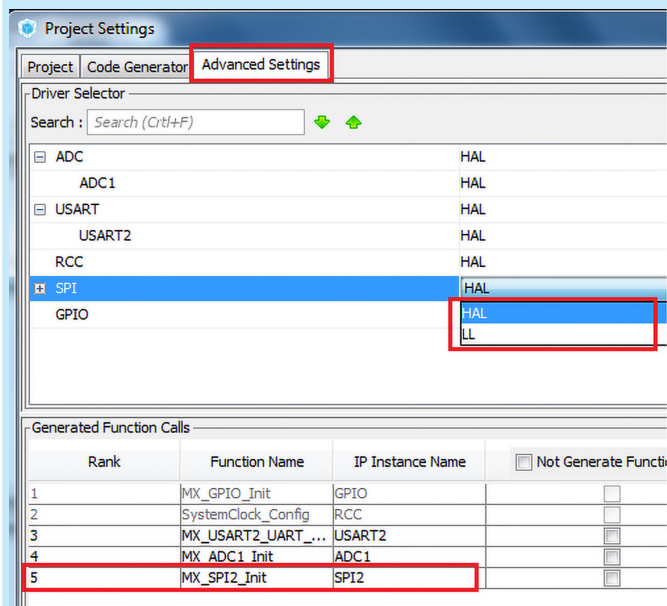
Rysunek 41. Zakładka Code Generator okna Project Settings



Rysunek 39. Alternatywne graficzne przedstawienie poboru prądu przez zdefiniowane kroki

W zasadniczej części procesu generowania kodu są kopiowane wybrane pliki biblioteczne do folderów Drivers/CMSIS, Drivers/STM32F4_HAL_driver oraz folderu Middleware, jeżeli w konfiguracji wybrano modułu programowe Middleware. Potem generowane są pliki inicjalizacyjne (.c/.h) odpowiadające wybranej konfiguracji i zapisywane w folderach *inc* i *src*.

Generator kodu jest konfigurowany w oknie ustawień projektu Project Settings (Project → Settings). W zakładce Project (rysunek 40) ustawiamy głównie nazwę głównego folderu aplikacji (src) i środowisko projektowe dla którego ma być wygenerowany kod. W naszym przypadku będzie to oczywiście TrueSTUDIO.



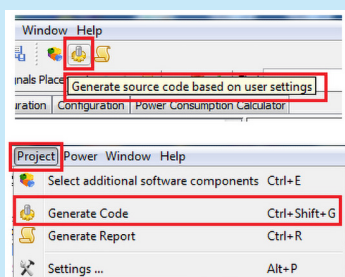
Rysunek 42. Zakładka Advanced Settings

W oknie Project Location musi być ścieżka dostępu do katalogu z przestrzenią roboczą Workspace. W przeciwnym przypadku nie będzie można otworzyć wygenerowanego projektu.

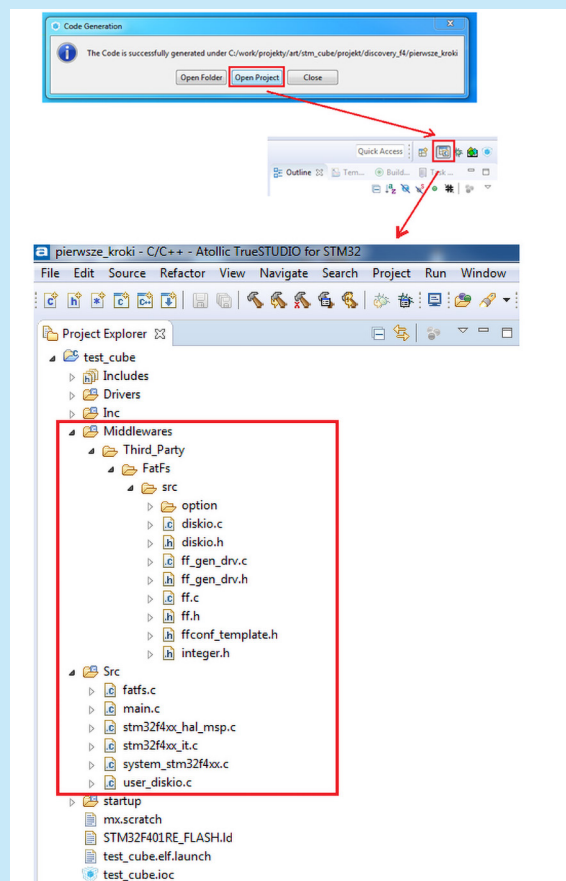
W zakładce Code Generator domyślnie jest ustawione kopiowanie z bibliotek tylko potrzebnych plików, oraz zachowywanie kodu użytkownika po rekonfiguracji peryferii i wygenerowaniu nowego kodu przez STM32CubeMX. Wymaga to zaznaczenia w kodzie sekcji, która nie może zostać usunięta przez znaczniki:

```
/* USER CODE BEGIN PVx */
i
/* USER CODE END PVx */
```

Zakładkę Code Generator pokazano na rysunku 41. Ostatnia zakładka Advanced Settings (rysunek 42) pozwala na wybranie rodzaju wygenerowanego drivera dla układu peryferyjnego. Może to być driver warstwy abstrakcji HAL lub driver najniższego poziomu LL



Rysunek 43. Uruchomienie generowania kodu wynikowego



Rysunek 44. Otwieranie wygenerowanego projektu

(LowLayer). W liście Generated Functions Call są wypisane nazwy funkcji inicjalizacyjnych oraz nazwy instancji (wcieleń).

Generowanie kodu na podstawie ustawień uruchamia się przez kliknięcie ikony Generate Source Code z paska narzędziowego wtyczki lub z menu Project → Generate Code (rysunek 43). Jeżeli CubeMX nie wykrył błędów w konfiguracji, to utworzy szkielet projektu dla Atollic TrueESTUDIO i umieści w nim pliki konfiguracyjne zgodne z naszymi ustawieniami. Po zakończeniu otwiera się okno z pytaniem, czy otworzyć wygenerowany projekt, czy tylko folder z projektem. Po kliknięciu na „open project” możemy przejść do edycji plików źródłowych. Trzeba tylko zmienić perspektywę na C/C++ jak to zostało pokazane na rysunku 44.

Tomasz Jabłoński, EP

REKLAMA

MEDIA ELEKTRONIKA PRAKTYCZNA

Od stycznia br. zmieniliśmy sposób dostarczania Czytelnikom EP materiałów dodatkowych dołączonych do numeru.

1. Wejść na stronę www.media.avt.pl
2. Zarejestruj się/zaloguj
3. Wybierz wydanie „Elektroniki Praktycznej”, które chcesz dodać do swojej biblioteki.
4. Odpowiedz na proste pytanie dotyczące bieżącego numeru.
5. Pobieraj pliki.

