PROJEKTOWANIE ENERGOOSZCZĘDNYCH SYSTEMÓW WBUDOWANYCH

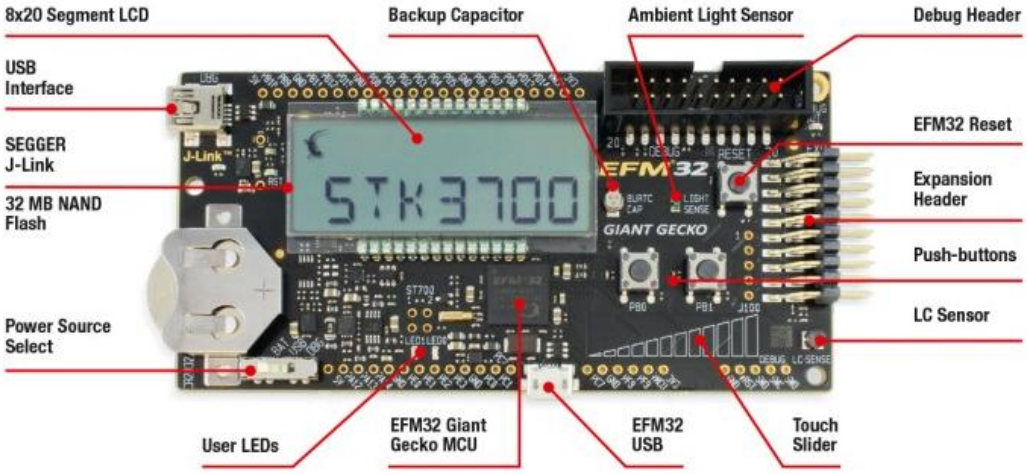
LABORATORIUM 1

PLATFORMA SPRZĘTOWA EFM32GG-STK3700 – PIERWSZE KROKI

KRAKÓW, 2016

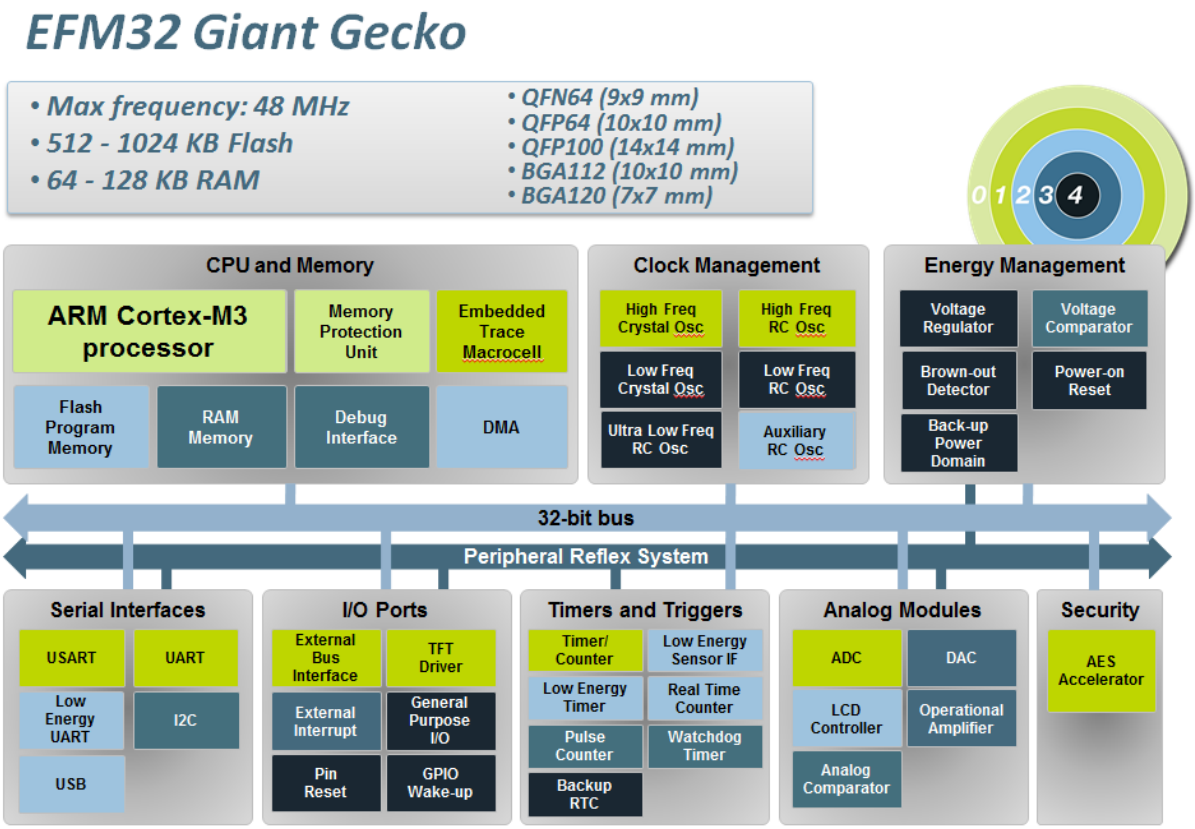
1. Wprowadzenie

W bloku ćwiczeń laboratoryjnych poświęconych nauce tworzenia energooszczędnych aplikacji na mikrokontrolery będziemy korzystać z platformy sprzętowej EFM32GG-STK3700 (Rysunek 1).



Rysunek 1. Płytka testowa STK3700

Platforma oparta jest o procesor EFM32GG990F1024 (rdzeń ARM Cortex-M3), którego peryferia przedstawione są na Rysunku 2.



Rysunek 2. Schemat mikrokontrolera EFM32GG

2. Zintegrowane środowisko programistyczne (IDE)

Do tworzenia oprogramowania na mikrokontrolery serii EFM32 można używać szeregu środowisk IDE: Keil, IAR, Rowley i wiele innych, w tym open source (Eclipse). Na zajęciach korzystać będziemy z oprogramowania wbudowanego w środowisko Simplicity Studio.

3. Simplicity Studio

Simplicity Studio jest aplikacją zawierającą w sobie wszelkie niezbędne programy, przykłady i dokumentację pomocną przy pracy z mikrokontrolerami EFM32. Aplikację można za darmo pobrać za strony www.silabs.com. Zawiera ona m.in.:
symulator czasu życia baterii, narządzie do programowania układu, narzędzie do konfiguracji pinów i generowania kodu startowego, narzędzie do obserwacji poboru prądu, dokumentację układów, dokumentację API, dokumentację płytek, noty aplikacyjne i przykłady.
Aplikacja umożliwia dodawanie kolejnych dostępnych aktualizacji podprogramów oraz danych technicznych układów EFM32.

4. **Główne okno środowiska Simplicity Studio** (wprowadzenie podczas zajęć)
Getting Started / Documentation / Compatible Tools / Resources

EnergyAware Battery, Flash Programmer, Hardware Configurator:
Narzędzia te znajdują się w zakładce "Compatible Tools" i służą kolejno do wyznaczania czasu pracy na zdanej baterii, programowania i zabezpieczania mikrokontrolera, konfigurowania pinów/portów mikrokontrolera.

Program Energy Profiler służy do analizy zużycia energii przez mikrokontroler w czasie rzeczywistym. Jest to bardzo dobre narzędzie do tzw. techniki energy debugging, czyli testowaniu kodu z punktu widzenia poboru prądu.

5. Uruchamianie pierwszego programu demo na STK3700 – mrugająca dioda

Po uruchomieniu środowiska wybieramy projekt STK3700 blink (mrugająca dioda) z Getting Started, w kolejnym oknie załączamy tryb pracy z Energy Profile i korelację z kodem programu. Zapoznajemy się z Energy Profile. Uruchomiony moduł zamykamy w prawym górnym rogu wybierając prawy przycisk myszki i opcję close.

6. Kompilacja i uruchamianie pierwszego programu na STK3700 – mrugająca dioda

W głównym oknie środowiska wybieramy z przykład STK3700_blink i tworzymy nowy projekt.
Otwiera się okno wbudowanego IDE z kodem programu.

```
/**************************************************************************
****//**
 * @brief  Main function
 *************************************************************************
******/
int main(void)
{
  /* Chip errata */
  CHIP_Init();

```

```
8.  /* If first word of user data page is non-zero, enable eA Profiler trace */
9.  BSP_TraceProfilerSetup();
10.
11. /* Setup SysTick Timer for 1 msec interrupts  */
12. if (SysTick_Config(CMU_ClockFreqGet(cmuClock_CORE) / 1000)) while (1) ;
13.
14. /* Initialize LED driver */
15. BSP_LedsInit();
16. BSP_LedSet(0);
17.
18. /* Infinite blink loop */
19. while (1)
20. {
21.   BSP_LedToggle(0);
22.   BSP_LedToggle(1);
23.   Delay(1000);
24. }
25. }
```

Po podstawowej inicjalizacji układu, inicjalizowane są diody LED0 i LED1 na płytce STK. następnie zaświecona zostaje dioda LED0. W nieskończonej pętli wykonuje się sekwencja: zmiana stanu diody LED0, zmiana stanu diody LED1, opóźnienie 1000 ms.

Do kompilacji i zbudowania naszego projektu służą przyciski (z menu Project) np. Build Project.

Mając zbudowany projekt mrugających diod należy wgrać nasz plik wsadowy do mikrokontrolera należy użyć przycisku Flash Programmer (z paska narzędzi).

Po wciśnięciu przycisku RESET na płytce STK, program zaczyna działać i możemy obserwować mrugające diody.

Środowisko IDE pozwala również na debuggowanie napisanego kodu. W tym celu po wgraniu pliku wsadowego należy uruchomić debuggowanie przyciskiem Debug (z menu Run). Korzystanie z debuggera jest możliwe za pomocą innych przycisków z menu Run oraz paska narzędzi.

Po zakończeniu debuggowania należy przyciskiem Disconnect odłączyć oprogramowanie od śledzenia stanu mikrokontrolera.

7. CMSIS API

Oprócz możliwości operowania tylko na rejestrach mikrokontrolera EFM32, dostępna jest również biblioteka API do obsługi wszystkich peryferii. Dokumentacja dostępna jest z poziomu Simplicity Studio.
Proszę zapoznać się ze strukturą dokumentacji biblioteki API w celu przyspieszenia pracy i lepszego zrozumienia materiału na kolejnych laboratoriach.
Zakładka Documentation "GECKO SDK Documentation".

8. Zadanie projektowe

Należy stworzyć program, który będzie w pętli mrugać diodą LED0/LED1 sekwencję "własne imię" w alfabecie Morse'a. Działanie programu zostanie zweryfikowane przez prowadzacego zajęcia.
Sprawozdanie w formie pdf zawierające kod wraz z opisem należy przesłać mailem.