



LABORATORIUM
Architektura systemów wbudowanych

Katedra Systemów Elektroniki Morskiej
Wydział Elektroniki Telekomunikacji i Informatyki
Politechnika Gdańska



Temat laboratorium:
Procesor sygnałowy

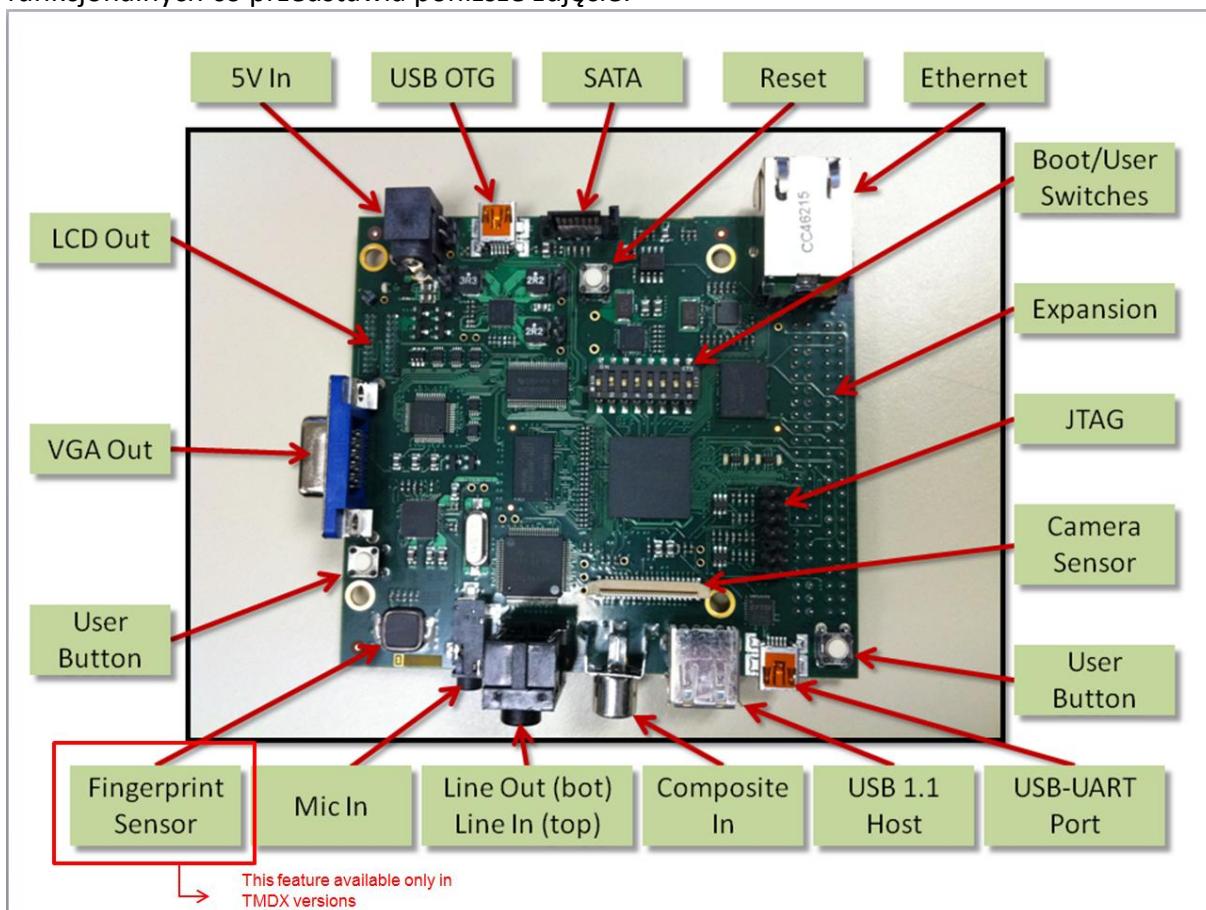


Opracował:
Jan H. Schmidt

2017

Wstęp.

Ćwiczenie laboratoryjne będzie realizowane z wykorzystaniem zestawu uruchomieniowego TMS320C6748 DSP Development Kit (LCDK) zawierający procesor sygnałowy firmy Texas Instruments. Płyta zestawu uruchomieniowego składa się z bloków funkcjonalnych co przedstawia poniższe zdjęcie:

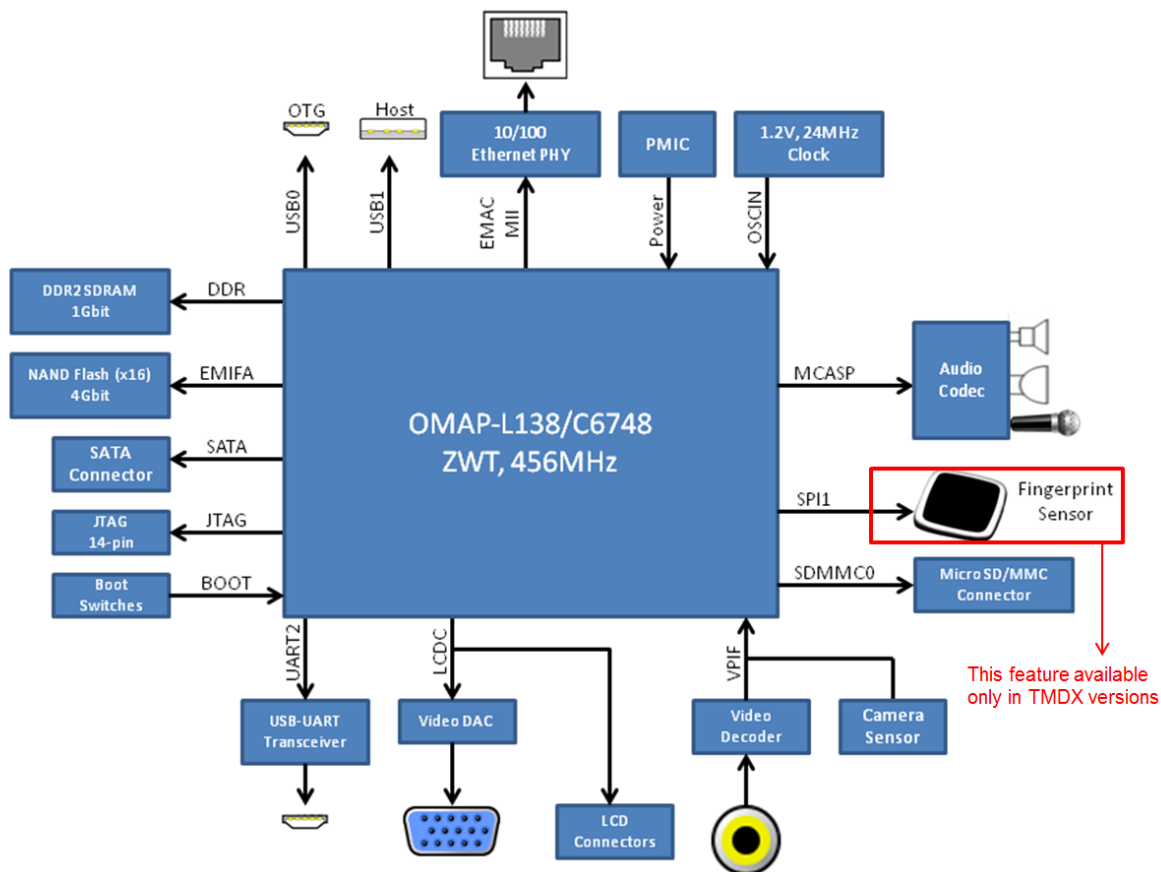


Rys. 1. Rozmieszczenie elementów, wyprowadzeń oraz bloków funkcyjnych na płycie zestawu uruchomieniowego

Płyta zawiera zastępujące elementy:

- Procesor sygnałowy TMS320C6748, taktowny zegarem 456 MHz ze zintegrowanym zegarem czasu rzeczywistego RTC
- Pamięć dynamiczna 128MB DDR2 SDRAM
- Pamięć Flash 128 MB 16-bit NAND
- Slot kart pamięci Micro SD/MMC
- Złącze mini-USB Serial (on-board serial to USB)
- Złącze Fast Ethernet (10/100 Mbps)
- Złącze USB Host (USB 1.1)
- Złącze mini-USB OTG (USB 2.0)
- Złącze SATA (3Gbps)
- Złącze VGA (15 pin D-SUB)

- Złącze LCD (standard złącz modułów Beagleboard XM)
- Złącze wejściowe Composite Video Input (RCA Jack)
- Złącze wejściowe Leopard Imaging Camera Sensor (36-pin ZIP connector)
- Złącza układu kodeka AUDIO (1 LINE IN & 1 LINE OUT & 1 MIC IN)
- Złącze 14-pin JTAG do podłączenia zewnętrznego emulatora interfejsu JTAG
- Złącza z wyprowadzoną magistralą sterującą, adresową i danych procesora do podłączania układów dodatkowych



Rys. 2. Najważniejsze wykorzystywane interfejsy zestawu uruchomieniowego

Głównym elementem zestawu uruchomieniowego jest 32-bitowy procesor sygnałowy TMS320C6748. Zawiera on rdzeń C674x, który jest zoptymalizowany do realizacji złożonych operacji zmiennoprzecinkowych z maksymalną wydajnością 2746 MFLOPS / 3648 MIPS. Obszerna i efektywna wewnętrzna pamięć procesora umożliwia uzyskanie maksymalnej wydajności i realizacji zadań czasu rzeczywistego. Procesor jest uzupełniony o liczne interfejsy które czynią go elastycznym elementem projektowanych systemów.

Wewnętrzną pamięcią procesora o rozmiarze i pamięć SDRAM służą do przechowywania programu oraz danych. Pamięć Flash jest używana do przechowywania programu służącego do uruchomienia procesora po podaniu zasilania.

Procesory sygnałowe są często używane w aplikacjach przetwarzających dźwięk dlatego na płycie umieszczony został układ kodeka (TLV320AIC3106). Kodek umożliwia kodowanie i dekodowanie, a jego działanie polega na przetwarzaniu sygnału z postaci analogowej na postać cyfrową dla późniejszej obróbki przez procesor i zamianę sygnału w postaci cyfrowej na postać analogową. Do komunikacji z układem kodeka wykorzystano interfejs McASP.

Praca z modułem uruchomieniowym

1. Ustawienia początkowe modułu uruchomieniowego.

Zanim przystąpimy do uruchomienia zintegrowanego środowiska programistycznego Code Composer Studio należy upewnić się, że ustawienia przełączników (DIP switchy) są zgodne z wymaganymi oraz podłączone są odpowiednie kable połączeniowe.

a. Dla przełączników wymagana jest następująca sekwencja:



Przełączniki oznaczone **2, 3, 4** ustawione mają być w pozycji **ON**
Przełączniki oznaczone **1, 5, 6, 7, 8** ustawione mają być w pozycji **OFF**

Znaczenie przełączników:

-wybór sposobu „bootowania” procesora.

Switch #	UART2	NAND 16	MMC/SD0
1	OFF	OFF	OFF
2	ON	ON	OFF
3	OFF	ON	OFF
4	ON	ON	ON

- przełączniki podłączone do określonych pinów procesora dla użycia przez użytkownika.

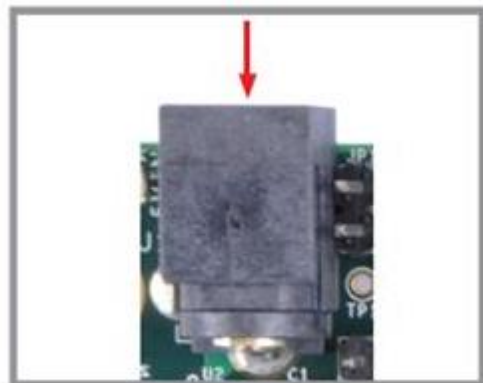
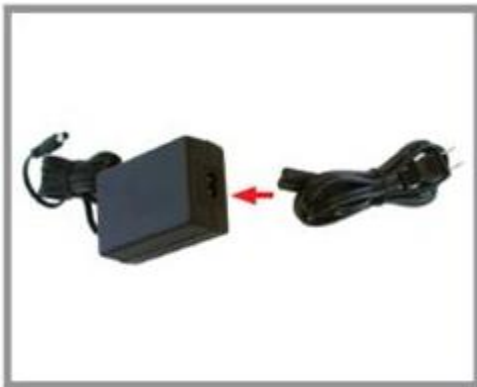
Switch #	Pin
5	GPIO0[1]
6	GPIO0[2]
7	GPIO0[3]
8	GPIO0[4]

Ustawianie:

ON odpowiada stanowi „low”

OFF odpowiada stanowi „high”

b. Połączyć zasilanie do modułu.



2. Zintegrowane środowisko programistyczne.

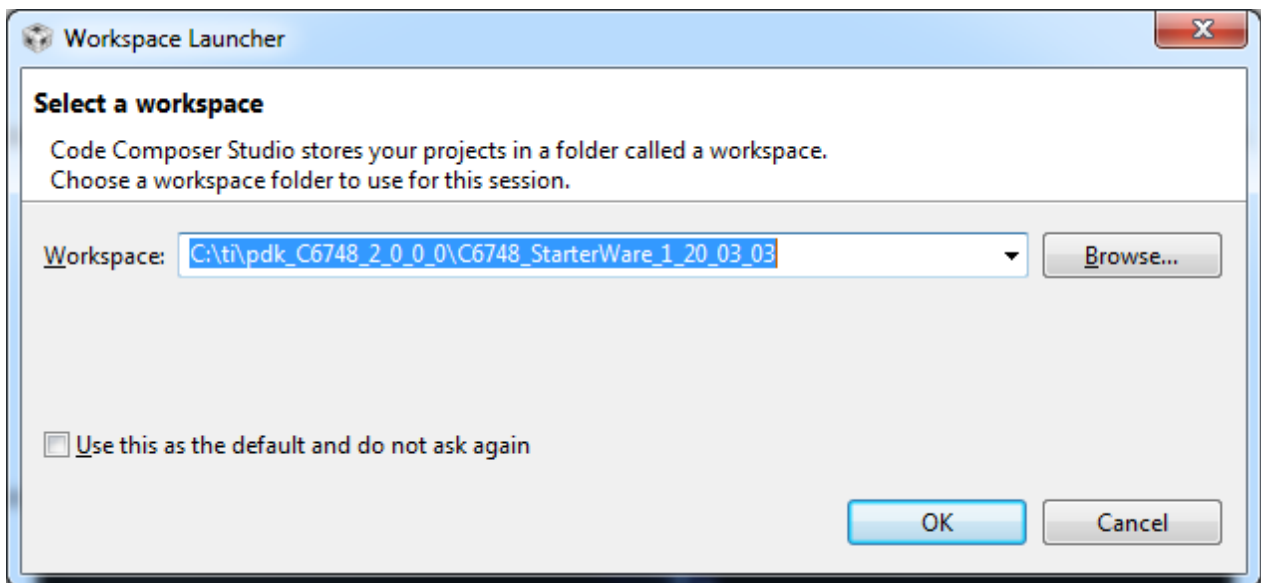
W celu uruchomienia zintegrowanego środowiska programistycznego należy skorzystać z ikony umieszczonej na pulpicie:



Rys.3. Ikona zintegrowanego środowiska programistycznego *Code Composer Studio*.

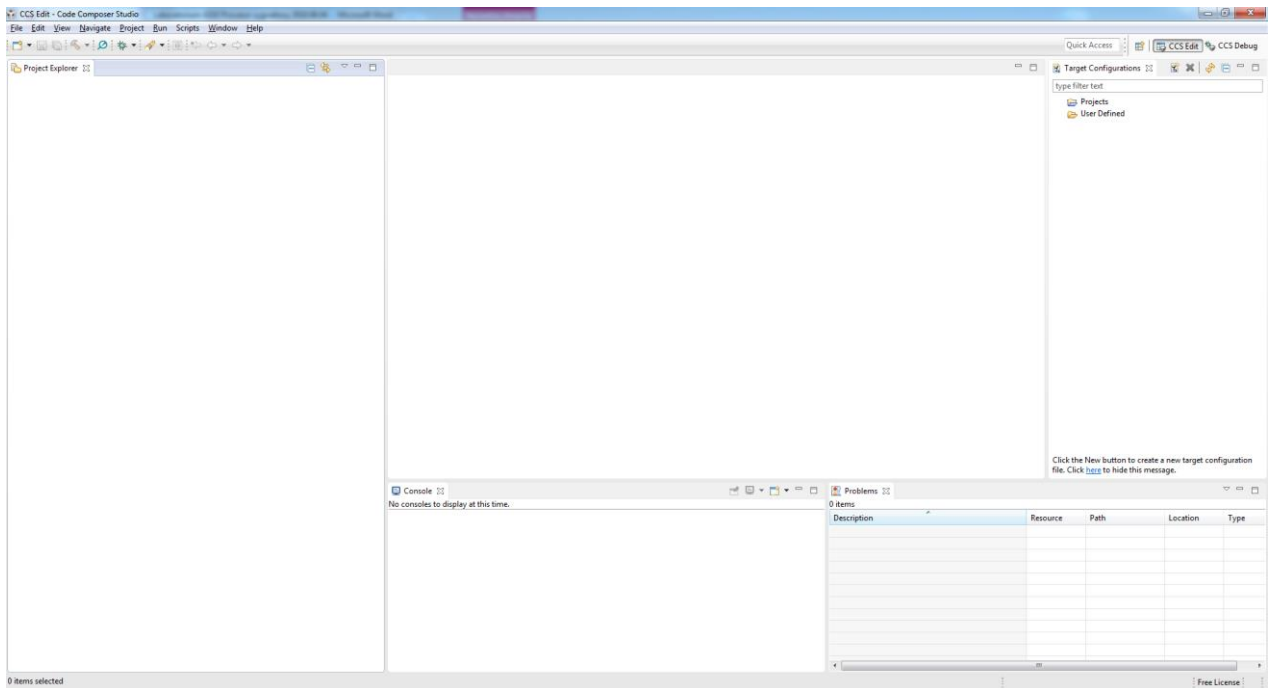
Podczas uruchamiania środowiska zostaniemy poproszeni o wybranie folderu roboczego projektu, który możemy wpisać lub wskazać za pomocą przycisku Browse co przedstawia poniższy rysunek. Opcja „Use this as the default and do not ask again” wybiera pracę z wykorzystaniem wspólnego folderu roboczego dla różnym projektów – należy unikać takiego rozwiązania.

Należy wprowadzić ścieżkę: **`C:\ti\pdk_C6748_2_0_0_0\C6748_StarterWare_1_20_03_03`**



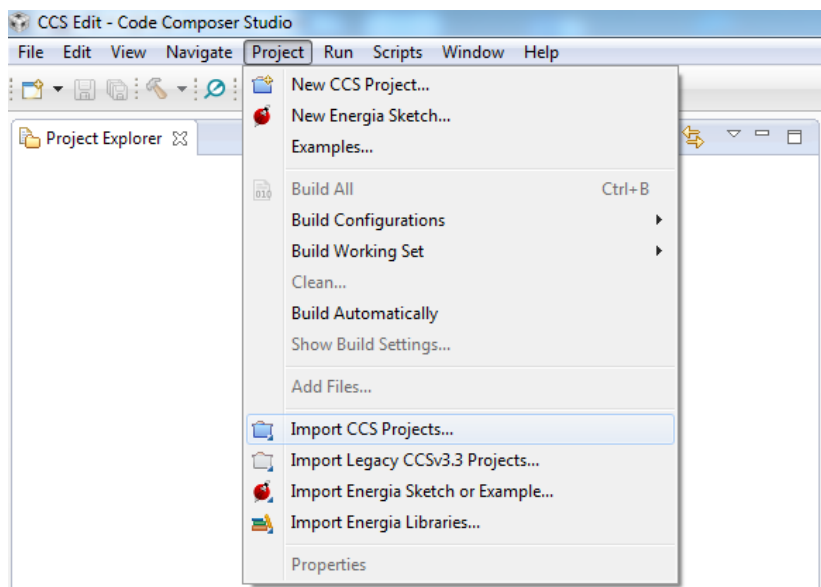
Rys.4. Okno edycyjne określania lokalizacji folderu roboczego projektu w środowisku CCS.

Okno główne po pierwszym uruchomieniu jest przedstawione na poniższym rysunku:

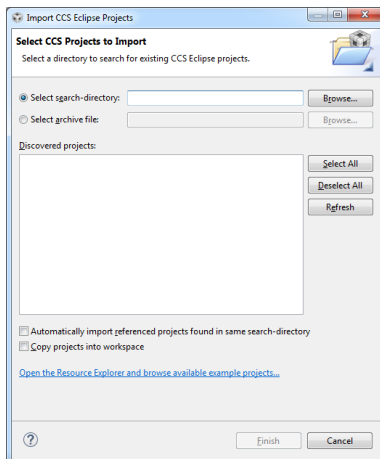


Rys.5. Okno główne środowiska Code Composer Studio 6.1.1

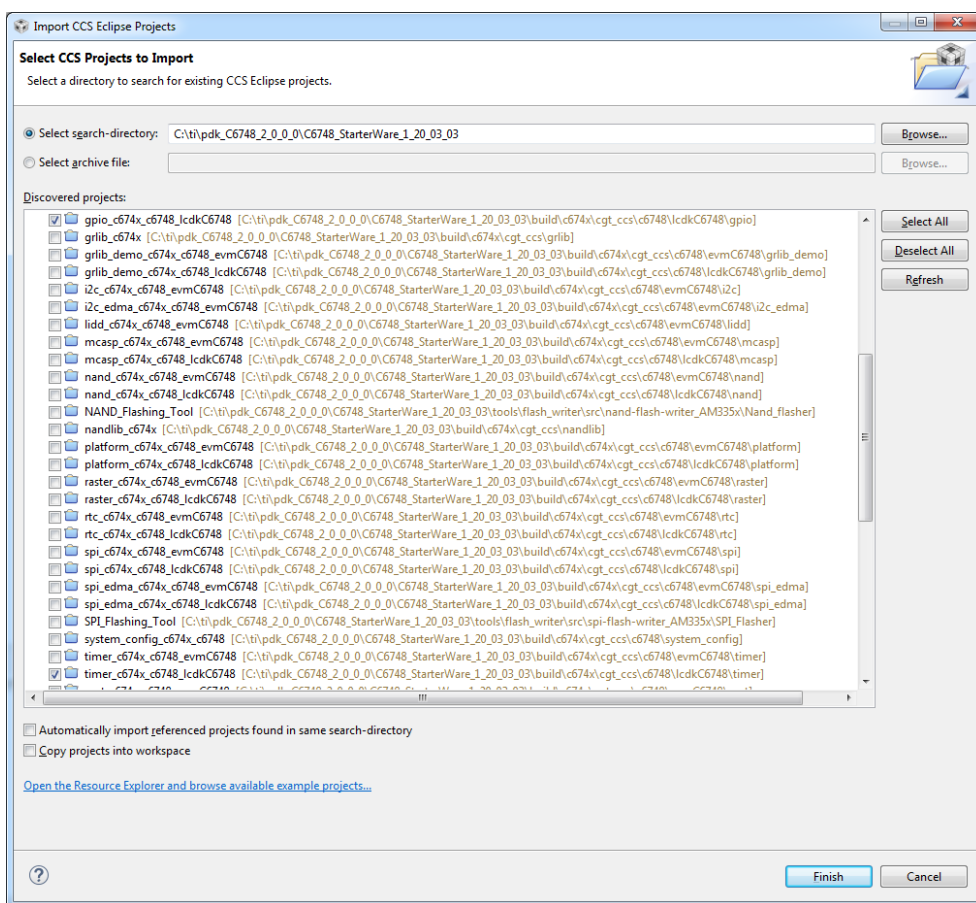
Aby odczytać wcześniej przygotowany projekt wybieramy opcję **Import CCS Projects...** z menu **Project**:

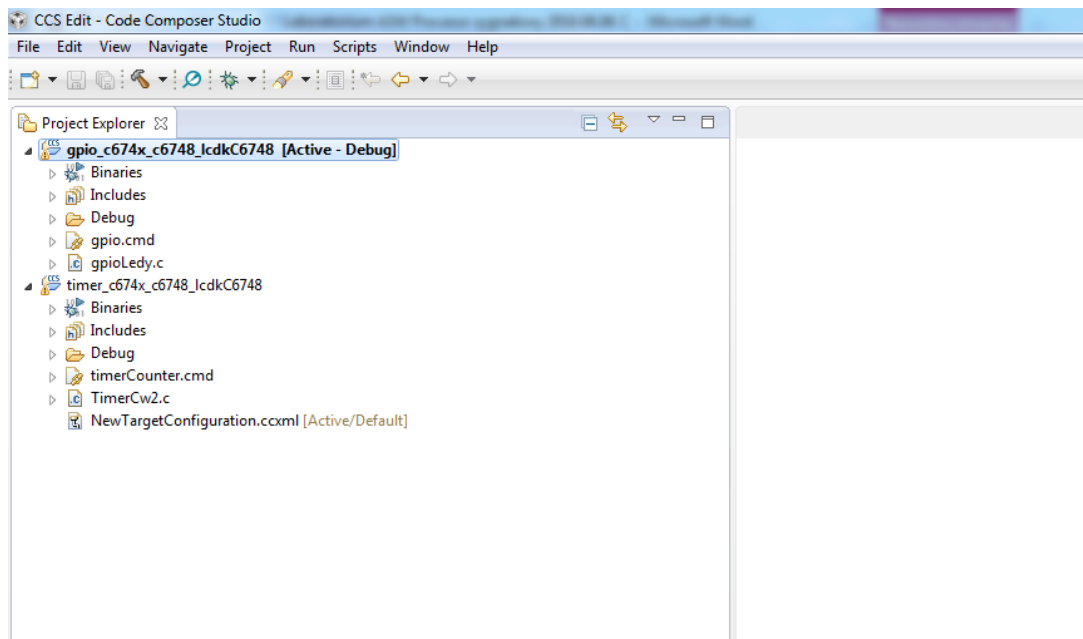


to spowoduje wyświetlenie okna:



Należy wybrać ścieżkę wyszukiwania projektów na **C:\ti\pdk_C6748_2_0_0_0\C6748_StarterWare_1_20_03_03**, a to spowoduje wyświetlenie listy wszystkich znalezionych projektów w tym katalogu. Projekty nas interesujące należy zaznaczyć i zostaną one dodane do listy projektów środowiska programistycznego co prezentują dwa następne rysunki.





Rys.6. Okno główne środowiska Code Composer Studio wraz z projektami z folderu roboczego.

Zadanie 1.

Korzystając z szablonu programu (**Projekt o nazwie *gpio_c674x_c6748_lcdkC6748***) zestawić program tak, aby realizował zadany program włączania/wyłączania diód LED w odpowiedzi na odpowiedni stan przełączników umieszczonych na płycie zestawu uruchomieniowego. Program włączania/wyłączania podaje prowadzący.

Kod programu stanowi rozliczenie zadania.

Informacje o przyciskach i diodach:

Button	Pin
S1	Reset
S2	GPIO2[4]
S3	GPIO2[5]

Rys.7. Podłączenie do procesora przycisków umieszczonych na płycie.

D4	GPIO6[13]	Signal is high
D5	GPIO6[12]	Signal is high
D6	GPIO2[12]	Signal is high
D7	GPIO0[9]	Signal is high

Rys.8. Podłączenie do procesora LEDów umieszczonych na płycie.

Dodatkowe informacje znajdziesz na stronie www.ti.com w następujących dokumentach:

- SPRUH79A - TMS320C6748 DSP Technical Reference Manual (Rev. A)
- SPRAB63B - OMAP-L132/L138, TMS320C6742/6/8 Pin Multiplexing Utility (Rev. B)

Zadanie 2.

Zmień szablon programu (**Projekt o nazwie *timer_c674x_c6748_lcdkC6748***) tak, aby procesor wykorzystując układ *Timer2* odmierzył czas **zadany przez prowadzącego w określonej liczbie powtórzeń**.

Moment startu odliczania i jego zakończenia należy zasygnalizować wykorzystując diodę LED.

Kod programu stanowi rozliczenie zadania.

Dodatkowe informacje znajdziesz na stronie www.ti.com w następujących dokumentach:

SPRUH79A - TMS320C6748 DSP Technical Reference Manual (Rev. A)

Zadanie 3.

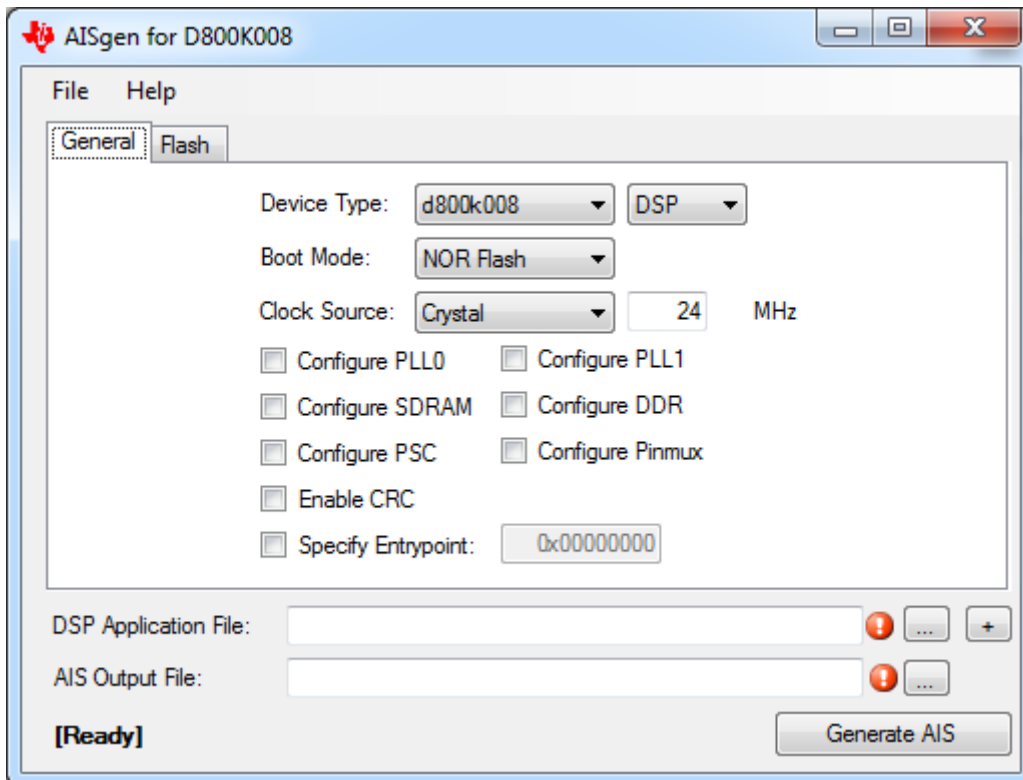
Na podstawie poprawnie działających programów z **zadania 1 i 2**, przygotować i wgrać do procesora program poprzez port szeregowy po włączeniu zasilania (resecie procesora).

Sposób postępowania:

- a. Przygotować plik binarny w formacie AIS (Application Image Script):

Wykorzystując plik wykonywalny z rozszerzeniem **.out** dokonujemy jego konwersji na plik binarny w formacie AIS odpowiedni dla *bootloader'a* procesora.

Należy uruchomić program **AISgen for D800K008** o lokalizacji c:\ti\AISgen_d800k008_Install_v1.13\AISgen for D800K008\AISgen_d800k008.exe co skutkuje pojawieniem się okna jak niżej:

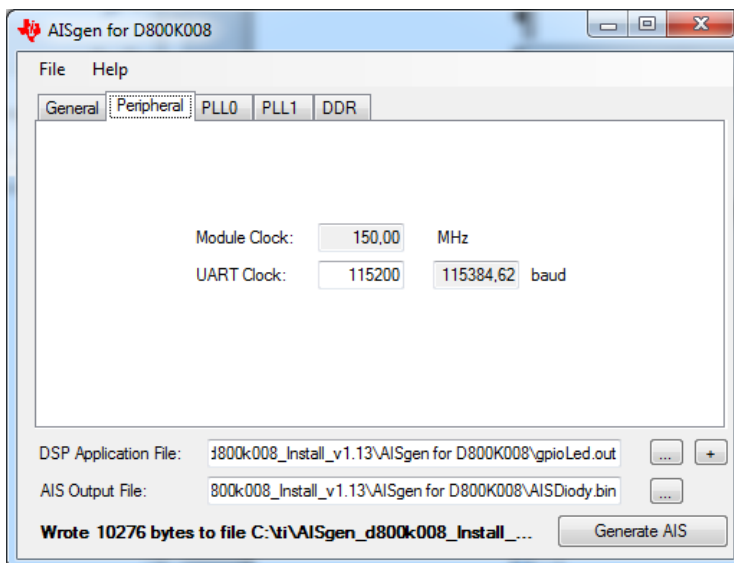
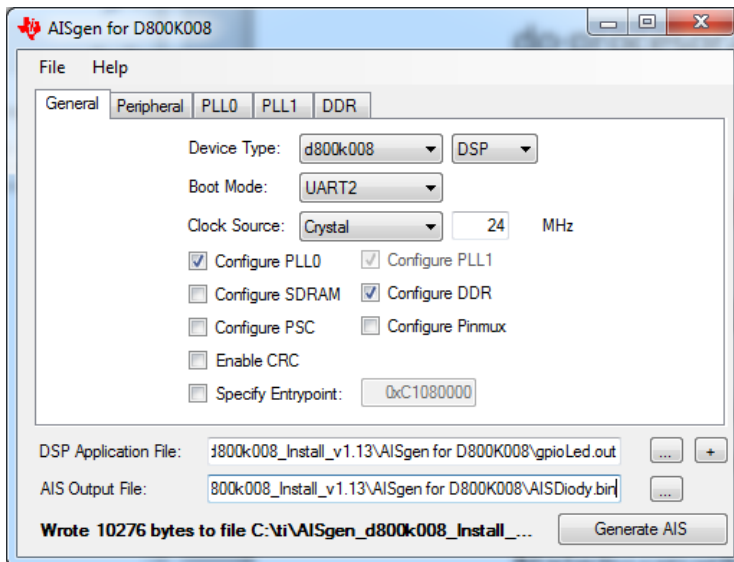


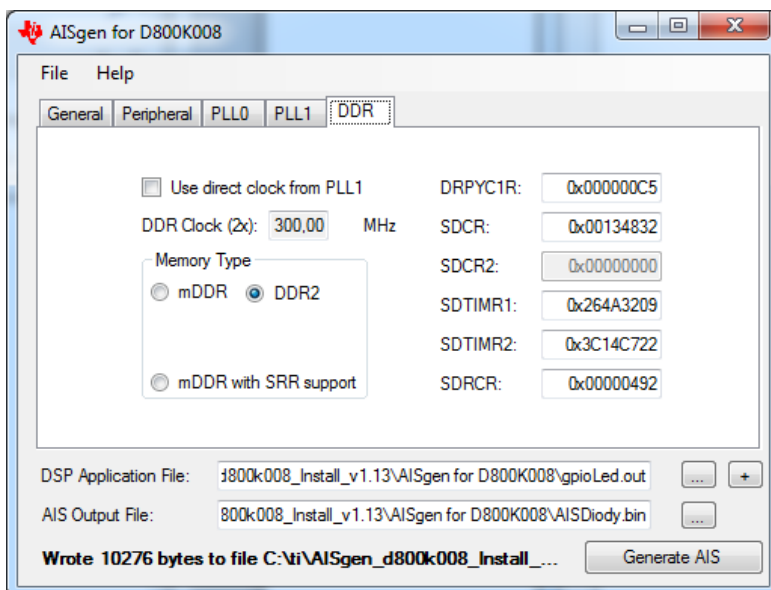
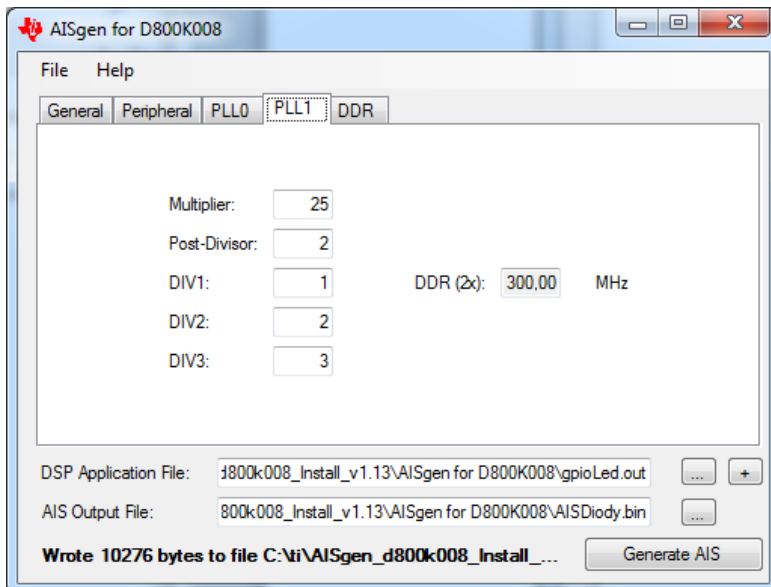
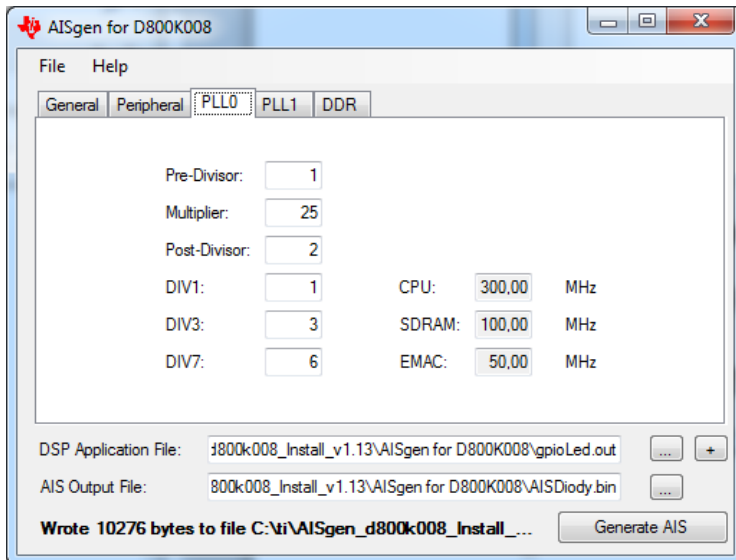
Za pomocą menu **File->Load Configuration** wgrać plik konfiguracyjny odpowiadający typowi używanego modułu uruchomieniowego (plik [C6748_LCDK_AISgen_Config.cfg](c6748_lcdk_aisgen_config.cfg)) o lokalizacji c:\ti\AISgen_d800k008_Install_v1.13\AISgen for D800K008\cfg_files

Po wybraniu trybu bootowania za pomocą portu szeregowego procesora *UART2*, nazwy pliku wykonywalnego i określeniu nazwy pliku binarnego - okna nastaw powinny zawierać parametry jak przedstawiają kolejne okna programu AISgen. Lokalizacja plików wykonywalnych:

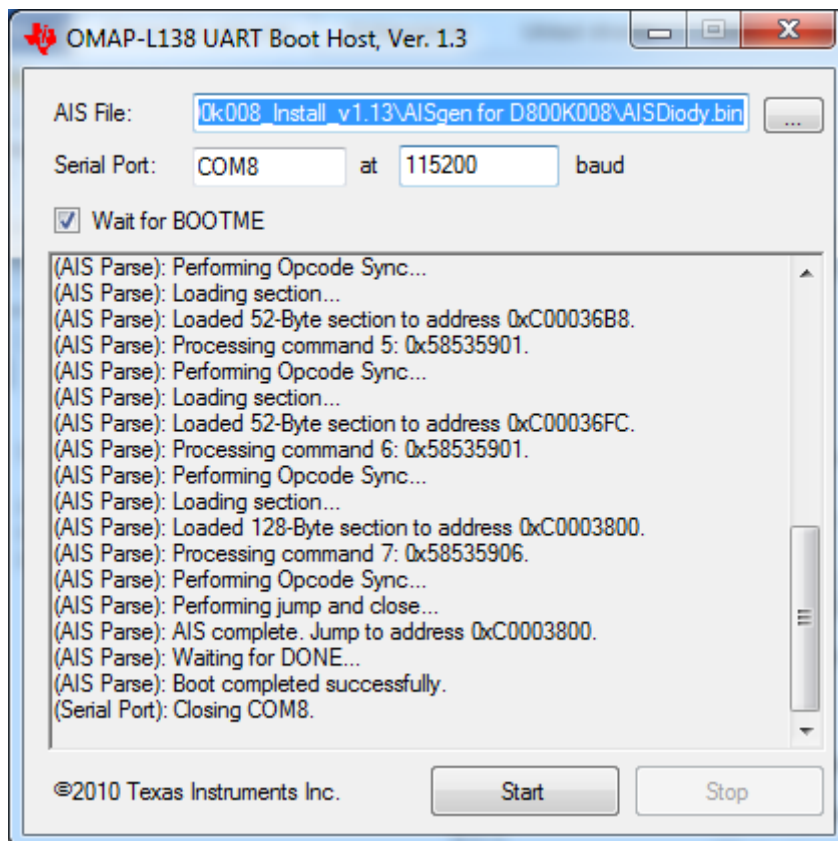
Zadanie 1 `c:\ti\pdk_C6748_2_0_0_0\C6748_StarterWare_1_20_03_03\build\c674x\cgt_ccs\c6748\cdkC6748\gpio\Debug\gpioLed.out`

Zadanie 2 `c:\ti\pdk_C6748_2_0_0_0\C6748_StarterWare_1_20_03_03\build\c674x\cgt_ccs\c6748\cdkC6748\timer\Debug\timerCounter.out`





- b. **Wyłączyć zasilanie modułu uruchomieniowego !!!**
- c. Ustawić przełączniki odpowiedzialne za wybór sposobu „bootowania” procesora w tryb UART2 (*patrz strona 4 instrukcji*)
- d. **Włączyć zasilanie modułu uruchomieniowego !!!**
- e. Dokonać ładowania programu w postaci pliku binarnego z pomocą programu ładującego UartHost o lokalizacji C:\ti\AISgen_d800k008_Install_v1.13\AISgen_for_D800K008\UartHost\



Po wyborze pliku do ładowania oraz portu szeregowego COMx na komputerze PC (*należy go ustalić w Menadźerze urządzeń systemu Windows*) wciskamy przycisk **Start**, a na module uruchomieniowym wciskamy przycisk **Reset** co spowoduje rozpoczęcie procedury ładowania programu.

Dodatkowe informacje znajdziesz na stronie www.ti.com w następujących dokumentach:

SPRUH79A [TMS320C6748 DSP Technical Reference Manual \(Rev. A\)](#)
 SPRAAT2F [Using the TMS320C6748/C6746/C6742 Bootloader \(Rev. F\)](#)

Zadanie 4.

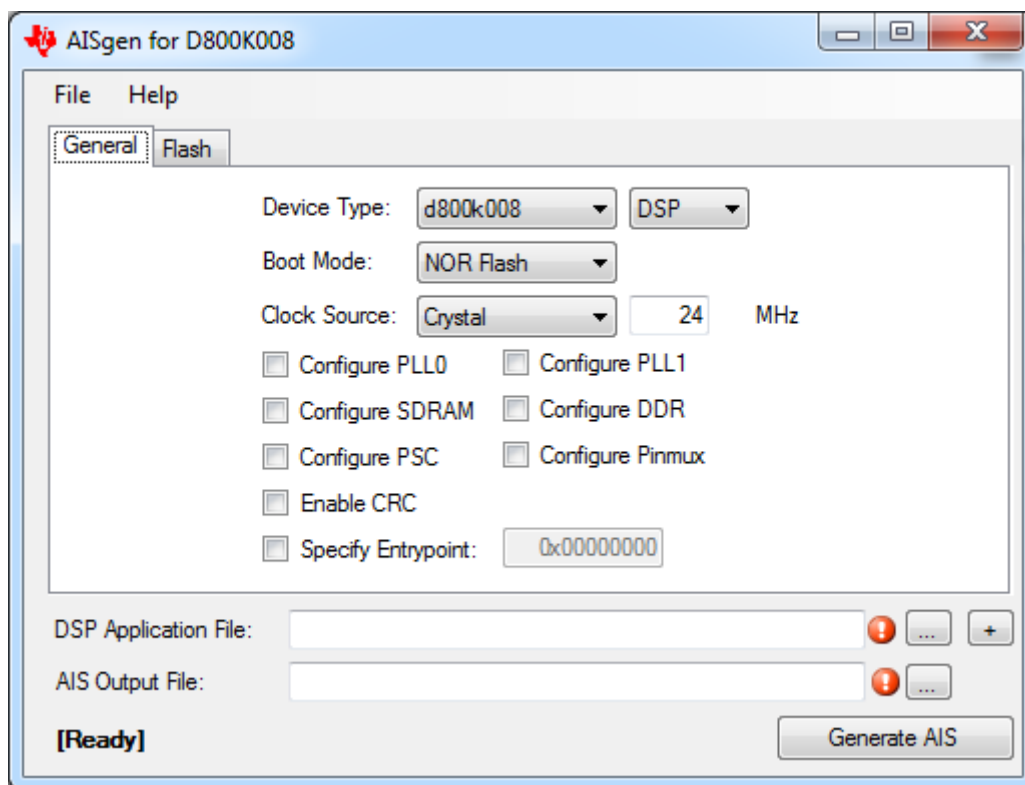
Dokonać „bootowanie” modułu z wykorzystaniem poprawnie działających programów z **zadania 1 i 2**, poprzez zaprogramowanie pamięci FLASH typu NAND.

Sposób postępowania:

- a. Przygotować plik binarny w formacie AIS (Application Image Script):

Wykorzystując plik wykonywalny z rozszerzeniem **.out** dokonujemy konwersji na plik binarny w formacie AIS, które jest akceptowalny przez *bootloader* procesora.

Należy uruchomić program **AISgen for D800K008** o lokalizacji c:\ti\AISgen_d800k008_install_v1.13\AISgen for D800K008\AISgen_d800k008.exe co skutkuje pojawieniem się okna jak niżej:

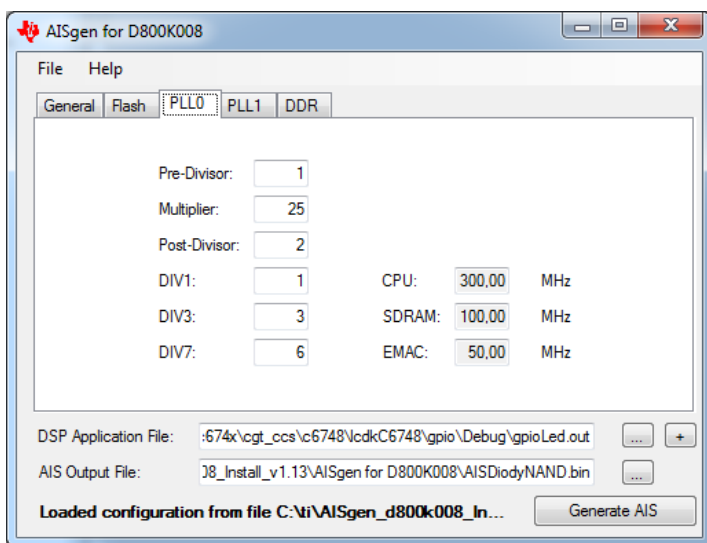
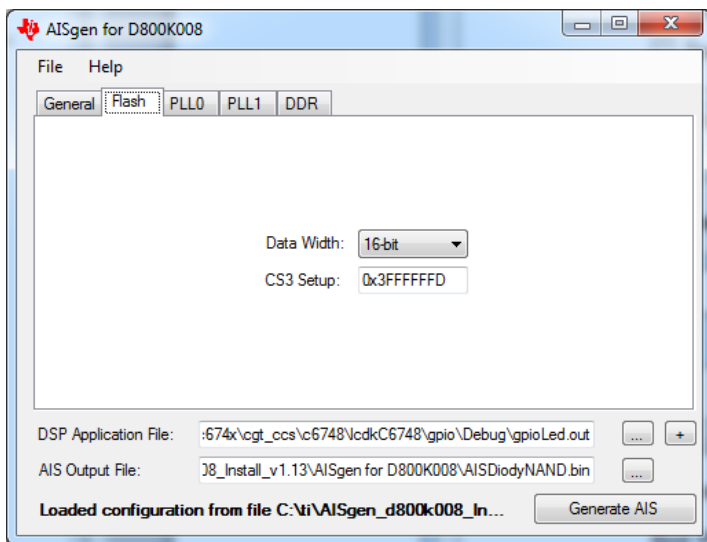
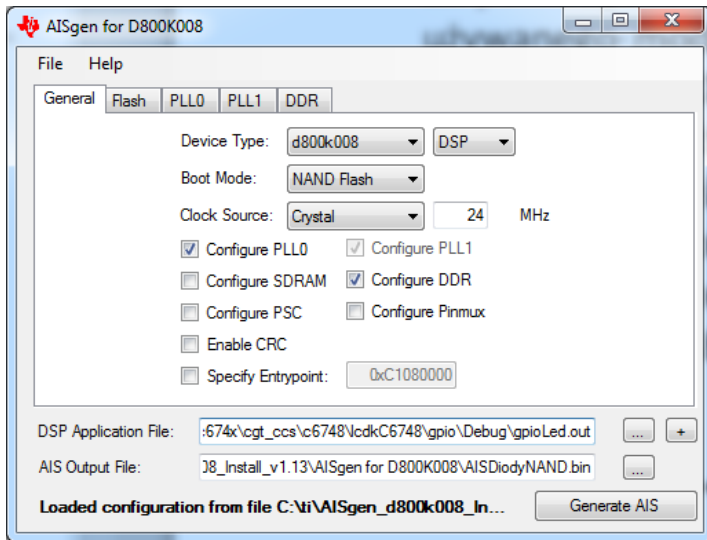


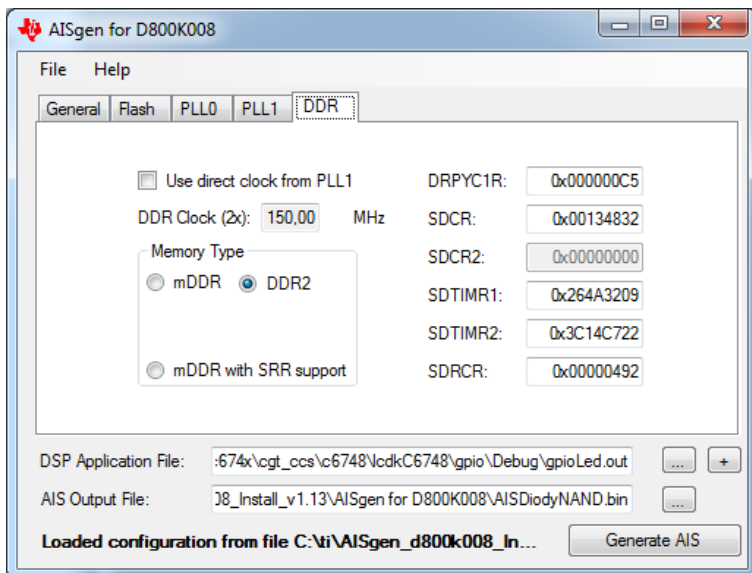
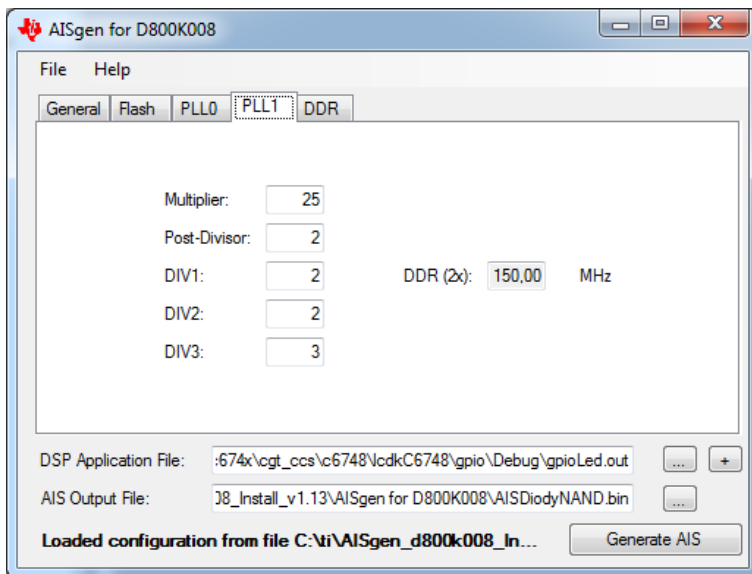
Za pomocą menu **File->Load Configuration** wgrać plik konfiguracyjny odpowiadający typowi używanego modułu uruchomieniowego (plik C6748_LCDK_AISGen_Config.cfg) o lokalizacji c:\ti\AISgen_d800k008_install_v1.13\AISgen for D800K008\cfg_files

Po wybraniu trybu bootowania z pamięci Flash NAND, podaniu nazwy pliku wykonywalnego do konwersji i określeniu nazwy pliku binarnego - okna nastaw powinny zawierać parametry jak przedstawiają kolejne okna programu AISgen. Lokalizacja plików wykonywalnych:

Zadanie 1 c:\ti\pdk_C6748_2_0_0_0\C6748_StarterWare_1_20_03_03\build\c674x\cgt_ccs\c6748\cdkC6748\gpio\Debug\gpioLed.out

Zadanie 2 `c:\ti\pdk_C6748_2_0_0_0\C6748_StarterWare_1_20_03_03\build\c674x\cgt_ccs\c6748\cdkC6748\timer\Debug\timerCounter.out`





b. Wyłączyć zasilanie modułu uruchomieniowego !!!

c. Ustawić przełączniki odpowiedzialne za wybór sposobu „bootowania” procesora w tryb UART2 (patrz strona 4 instrukcji)

d. Włączyć zasilanie modułu uruchomieniowego !!!

e. Uruchom wiersz poleceń sesji DOSowej na komputerze PC (cmd.exe) :

f. Zmienić ścieżkę pracy na:

`cd c:\ti\c6sdk_02_00_00_00\tools\OMAP-L138_FlashAndBootUtils_2_36\OMAP-L138\GNU`

g. Dokonać kasowania pamięci Flash NAND poprzez wydanie komendy:

```
sfh_OMAP-L138.exe -targettype C6748_LCDK -flashtype NAND -p COMx -erase
```

gdzie COMx to port na komputerze PC, który należy ustalić w Menadżerze urządzeń systemu Windows.

h. Wygenerowany wcześniej plik binarny przegrać do wcześniej określonego katalogu
c:\ti\c6sdk_02_00_00_00\tools\OMAP-L138_FlashAndBootUtils_2_36\OMAP-L138\GNU

i. Dokonać programowania pamięci Flash NAND poprzez wydanie komendy:

```
sfh_OMAP-L138.exe -targettype C6748_LCDK -flashtype NAND -p COMx -  
flash_noubl AISDiodyNAND.bin
```

j. Teraz programator (program [sfh_OMAP-L138](#)) powinien oczekiwać na komendę
BOOT ME od procesora, co można wykonać poprzez wciśnięcie przycisku RESET.

Należy poczekać aż zakończy się programowanie !!!.

k. **Wyłączyć zasilanie modułu uruchomieniowego !!!**

l. Ustawić przełączniki odpowiedzialne za wybór sposobu „bootowania” procesora
w trybie NAND 16 ([patrz strona 4 instrukcji](#))

m. **Włączyć zasilanie modułu uruchomieniowego !!!**

Sprawdzić czy moduł pracuje zgodnie z przygotowanym programem.

Dodatkowe informacje znajdziesz na stronie www.ti.com w następujących dokumentach:

SPRUH79A [TMS320C6748 DSP Technical Reference Manual \(Rev. A\)](#)
SPRAAT2F [Using the TMS320C6748/C6746/C6742 Bootloader \(Rev. F\)](#)

Zadanie 5.

Dokonać porównania czasów obliczeń zaimplementowanego algorytmu FFT (*Fast Fourier Transform*) - w języku C. Czasy obliczeń są wyznaczone dla różnych długości oraz różnych typów pamięci w której są przechowywane program i dane.

Sposób postępowania:

Należy wykorzystać projekt o nazwie **demo_c674x_c6748_lcdkC6748**. W pliku programu **CwiczenieFft.c** zdefiniowana została wartość **N**, która oznacza długość transformacji. Podczas kolejnych obliczeń należy zastosować **N** ze zbioru {64, 128, 256, 512, 1024, 2048, 4096}. Pierwszą serię obliczeń należy wykonać przy wykorzystaniu pamięci wewnętrznej procesora (**L2**), a drugą dla przypadku użycia zewnętrznej pamięci modułu uruchomieniowego (**DDR2**). Zmiany typu używanej pamięci należy dokonywać w pliku konfiguracyjnym linkera **demo.cmd**, tj. w części SECTIONS (> **L2** lub > **DDR2** dla wszystkich występujących tam sekcji).

Rozliczenie zadania stanowi tabela porównawcza.