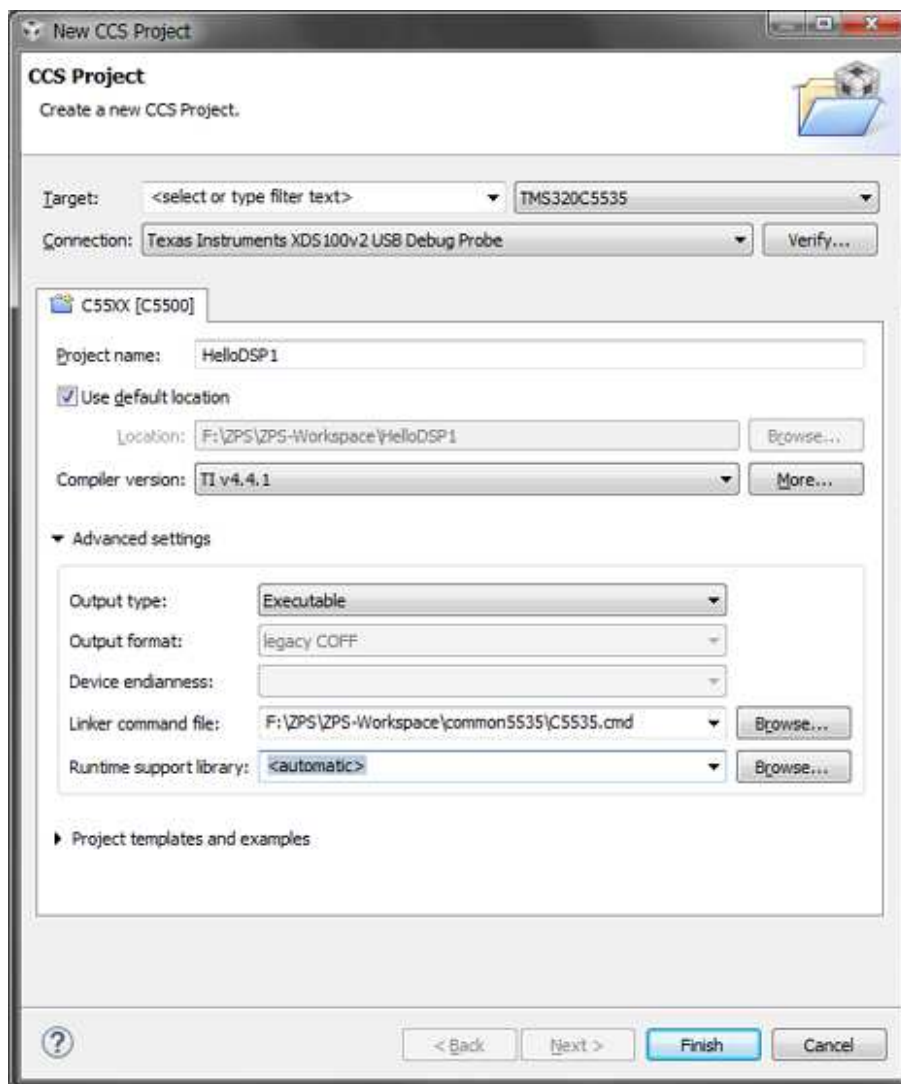


KONFIGURACJA NOWEGO PROJEKTU W CODE COMPOSER STUDIO

Tworzenie nowego projektu

- Podłączyć płytkę DSP (duży wtyk USB na płytce) do portu USB w komputerze.
- Uruchomić Code Composer Studio.
- W oknie *Workspace Launcher* podać ścieżkę do katalogu roboczego, w którym będzie zapisany projekt. Okno można też wywołać z menu: *File > Switch Workspace > Other*.
- Z menu *File* wybrać *New > CCS Project*.
- W polu *Target* rozwinąć listę z prawej strony i wybrać: *TMS320C5535*.
- W polu *Connection* rozwinąć listę i wybrać: *Texas Instruments XDS 100v2 USB Debug Probe*.
- Warto wcisnąć przycisk *Verify* i upewnić się że płytka „odpowiada”.
- W polu *Project name* wpisać nazwę projektu.
- Wybrać typ projektu: *Empty project*
- Kliknąć *Finish*.

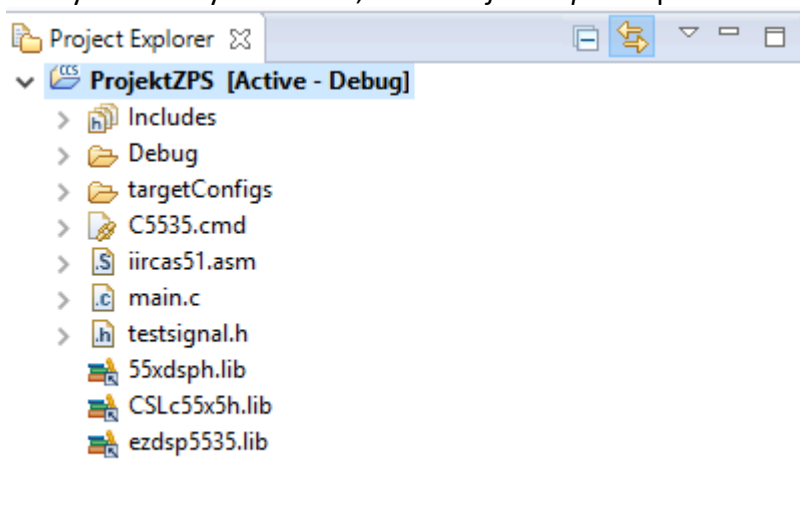


Instalacja plików do projektu

- Pobrać ze strony Katedry *Materiały pomocnicze*:
<http://www.multimed.org/student/materialy.html#zpsp>
plik opisany jako *Pliki potrzebne do utworzenia projektu CCS*
- Wszystkie pliki, które są „luzem” w katalogu roboczym należy przenieść do katalogu projektu (nadpisać istniejące pliki).
- W programie CCS w oknie *Project Explorer* rozwinąć drzewko projektu i upewnić się, że przeniesione pliki są tam widoczne.

Konfigurowanie projektu do kompilacji na DSP 5535

- W Eksploratorze Plików otworzyć katalog *lib* wewnątrz katalogu roboczego. Zaznaczyć wszystkie pliki *.lib* i przeciągnąć je myszką na nazwę projektu w oknie *Project Explorer* w CCS. Zaznaczyć opcję *Link to files* i zatwierdzić. Alternatywnie, można kliknąć prawym przyciskiem myszy na nazwie projektu i wybrać *Add files*.
- Kliknąć prawym przyciskiem myszy na nazwie projektu, wybrać *Properties*.
- Wybrać *Build > C5500 Compiler > Processor Options*, zmienić *memory model* z *large* na **huge**.
- Z *Advanced options* wybrać *Runtime Model Options*, zmienić *Specify type size to hold results of pointer math (--ptrdiff_size)* na **32**.
- Wybrać *Build > C5500 Compiler > Include Options*. W górnym polu (*Add dir to #include search path*): wcisnąć przycisk z zielonym plusem (*Add*). Dodać pozycję wpisując z klawiatury:
\${WORKSPACE_LOC}/include
- W tym momencie projekt jest skonfigurowany do pracy w trybie *Debug*. Jeżeli jest potrzeba wykorzystania konfiguracji *Release*, należy wybrać ją z listy *Configuration* i powtórzyć wszystkie kroki. Zatwierdzić przyciskiem *OK*.
- Po wykonaniu tych kroków, okno *Project Explorer* powinno wyglądać następująco:



Kompilacja projektu

- Upewnić się że projekt jest wybrany (Active) w panelu Project Explorer.
- Kliknąć przycisk z „młotkiem” na pasku narzędziowym.
- Upewnić się że nie ma błędów kompilacji – powinien być widoczny komunikat typu *'Finished building target: <nazwa>.out'*.
- Każda zmiana w kodzie wymaga jego ponownego skompilowania.

Uruchomienie projektu w trybie Debug

- Kliknąć ikonę z „pluską” na pasku narzędziowym, albo wcisnąć klawisz F11.
- Jeżeli projekt uruchomi się pomyślnie, program przełączy się do trybu debugowania. Wykonywanie programu zatrzyma się na punkcie startowym (zwykle nagłówek funkcji *main*). Jest to widoczne w postaci niebieskiej strzałki na marginesie edytora kodu.
- Uruchomienie programu nastąpi po wciśnięciu przycisku *Resume* (żółto-zielona ikona *play/pause*), lub po naciśnięciu klawisza F8.
- W tym momencie program powinien działać. Ewentualne błędy zostaną wyświetlone w programie.
- Wyjście z trybu debugowania – przycisk *Terminate* (ikona z czerwonym kwadratem) lub Control+F2. Uwaga: program na DSP nadal się wykonuje!
- UWAGA: czasami debugger wyświetli błąd *„Can't find a source file at...”*. Należy ten błąd zignorować, zamykając zakładkę z błędem i uruchamiając dalej program.

Debugowanie programu

- W edytorze można ustawić punkty przerwania (*breakpoint*) za pomocą dwukrotnego kliknięcia na marginesie z lewej strony edytora. Breakpoint jest zaznaczony niebieską kropką.
- Breakpointy można też ustawiać w trybie debugowania, nawet w trakcie działania programu! Skróty klawiszowe: Control+Shift+B.
- Program zatrzyma się po osiągnięciu breakpointa, PRZED wykonaniem linii, w której jest postawiony.
- Przesuwając kursor nad dowolną zmienną w edytorze można odczytać jej wartość.
- Zakładka *Variables* w górnej części okna programu pokazuje zdefiniowane lokalne zmienne i ich wartości, które można modyfikować.
- Zaznaczenie wyrażenia w edytorze i wybranie z menu kontekstowego opcji *Add watch expression* powoduje dodanie wyrażenia do zakładki *Expressions*.
- Przycisk *Step Into* (F5) powoduje przejście do kolejnej instrukcji, a jeżeli bieżącą instrukcją jest wywołanie funkcji, debugger wchodzi do wnętrza tej funkcji.
- Przycisk *Step Over* (F6) – jw., ale nie wchodzi do wnętrza funkcji.
- Przycisk *Step Return* (F7) wychodzi z funkcji, do której weszliśmy przez *Step Into*.
- Przycisk *Resume* (F8) wznowia działanie programu.

Wyświetlanie zawartości bufora

Założmy, że mamy bufor próbek o rozmiarze 2400:

```
int bufor[2400];
```

Chcemy wykreślić jego zawartość. Zatrzymujemy wykonywanie programu na breakpointie. W zakładce *Variables* lub *Expressions* klikamy prawym przyciskiem myszy na zmiennej *bufor* i wybieramy *Graph*. Okno bufora powinno pokazać się na dole okna.

Następnie należy otworzyć okno *Show The Graph Properties* klikając na ikonę paska narzędziowego wykresu i ustawić parametry jak poniżej.

- *Acquisition Buffer Size* – rozmiar bufora (tu 2400)
- *Dsp Data Type* – wybrać typ, np. *16 bit signed integer*
- *Sampling Rate Hz* – wpisać częstotliwość próbkowania, np. 48000
- *Start Address* – tu powinna być już automatycznie wpisany wskaźnik do bufora
- *Display Data Size* – należy wpisać rozmiar bufora (2400) lub mniej
- *Time Display Unit* – ustawić jednostkę osi czasu wg uznania.

Bufor zostanie uaktualniony za każdym razem gdy program zatrzyma się na breakpointie.

Property	Value
▲ Data Properties	
Acquisition Buffer Size	2400
Dsp Data Type	16 bit signed integer
Index Increment	1
Q_Value	0
Sampling Rate Hz	48000
Start Address	bufor
▲ Display Properties	
Auto Scale	<input checked="" type="checkbox"/> true
Axis Display	<input checked="" type="checkbox"/> true
Data Plot Style	Line
Display Data Size	2400
Grid Style	No Grid
Magnitude Display Scale	Linear
Time Display Unit	ms
Use Dc Value For Graph	<input type="checkbox"/> false

Można również wyświetlić FFT zawartości bufora. Należy z menu *Tools* wybrać *Graph > FFT Magnitude*. W oknie należy ustawić opcje jak powyżej (należy ręcznie wpisać *bufor* jako *Start Address*) oraz dodatkowo podać rozmiar FFT (zmieniając *FFT Order*) i typ okna.

Property	Value
Data Properties	
Acquisition Buffer Size	2400
Dsp Data Type	16 bit signed integer
Index Increment	1
Q_Value	0
Sampling Rate Hz	48000
Signal Type	Real
Start Address	bufor
Display Properties	
Auto Scale	<input checked="" type="checkbox"/> true
Axis Display	<input checked="" type="checkbox"/> true
Data Plot Style	Line
Frequency Display Unit	Hz
Grid Style	No Grid
Magnitude Display Scale	Linear
FFT	
FFT Frame Size	2048
FFT Order	11
FFT Window Function	Hamming

Konfiguracja projektu do obliczania FFT

Biblioteka DSPLIB zawiera procedury do obliczania FFT, dla danych rzeczywistych i zespolonych, 16- i 32-bitowych, do rozmiaru 1024 włącznie (dla sygnałów rzeczywistych: do 2048). Opis tych funkcji zawarty jest w dokumentacji DSPLIB. Korzystanie z tych funkcji wymaga jednak odpowiedniego skonfigurowania obszarów pamięci, w których będą zapisywane dane.

Konfigurację zapisuje się w pliku `.cmd` linkera. Poniższy przykład dotyczy konfiguracji zapisanej w pliku `C5535.cmd`, dostarczanej z szablonem projektu. Do poprawnego działania FFT, należy w bloku definicji sekcji (*SECTIONS*) dodać następujące wpisy:

```
.fftcode > SARAM0
.data:twiddle > SARAM1, align(2048)
.input > DARAM0, align(4)
```

Szczególnie istotny jest wpis dla `.data.twiddle`. Jeżeli go nie będzie, program skompiluje się i uruchomi, ale otrzymamy nieprawidłowe wyniki.

Definicja sekcji `.input` dotyczy danych, które będą przetwarzane przez FFT. Tablica próbek, która będzie podana do funkcji obliczającej FFT, musi zostać przydzielona do tej sekcji. W kodzie programu, przed zadeklarowaniem bufora próbek (koniecznie w części przed funkcją `main`), należy podać dyrektywę `DATA_SECTION`. Przykładowo dla tablicy o nazwie `dane`:

```
#pragma DATA_SECTION (bufor_probek, ".input")
DATA bufor_probek[N];
```