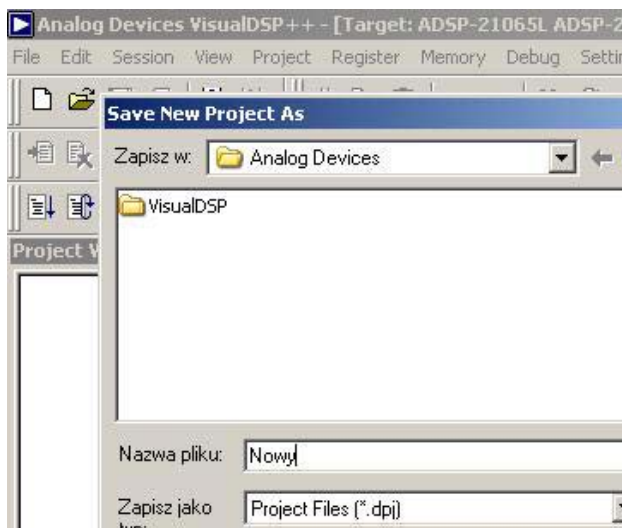


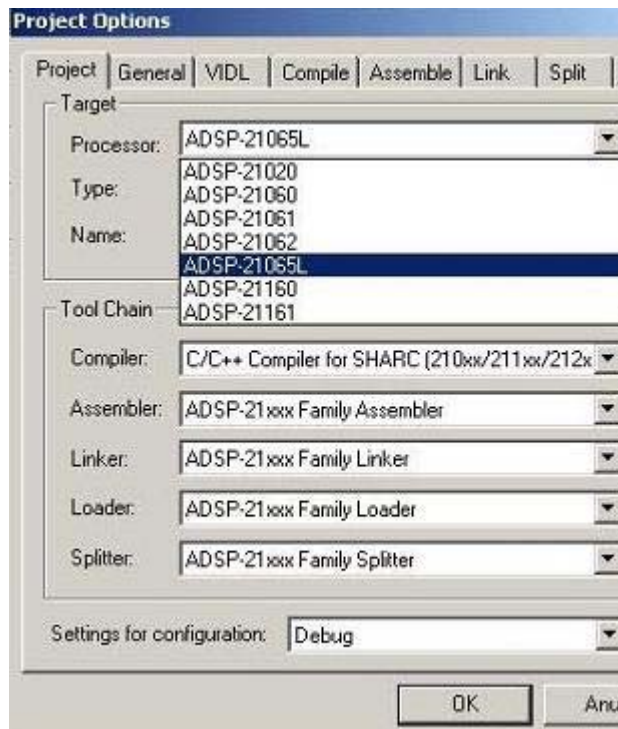
## ❑ Otwieramy nowy projekt



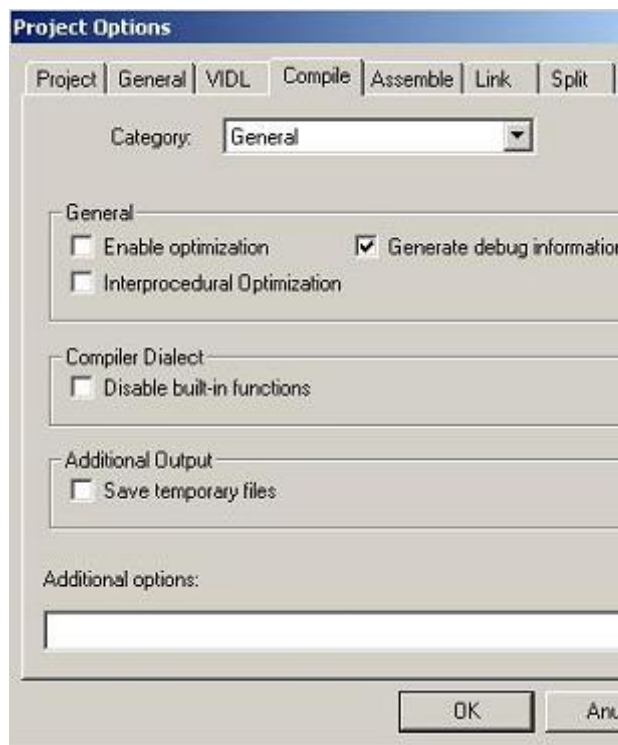
Z paska wybieramy **Project/New...**



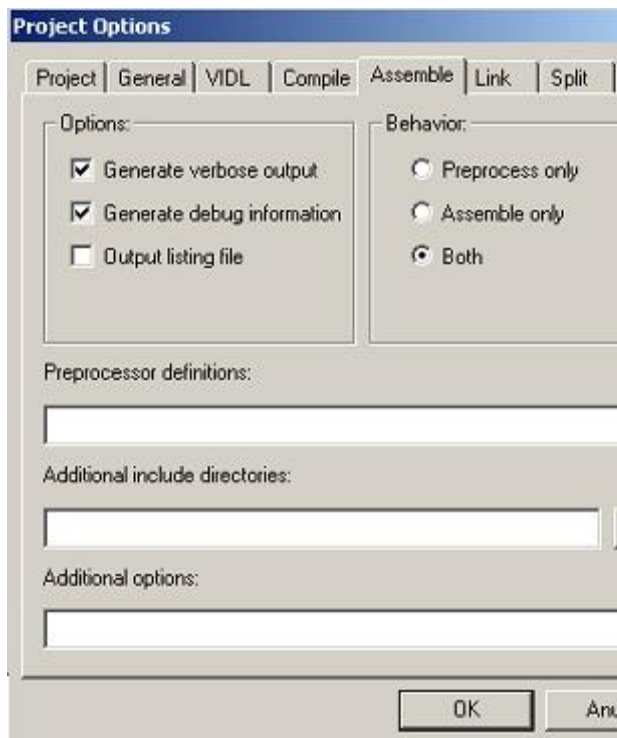
Zapisujemy jako „**Nowy**”



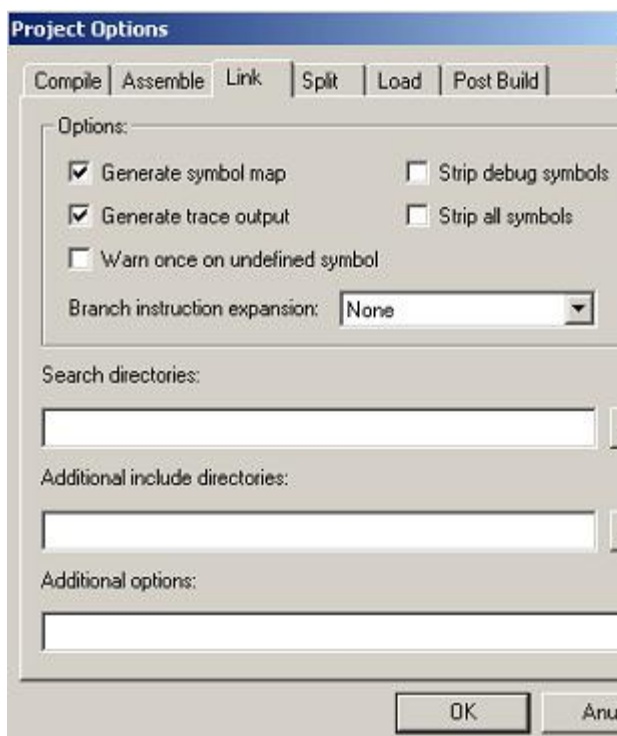
wybieramy **rodzaj procesora**



upewniamy się czy opcja **Generate debug information** jest zaznaczona

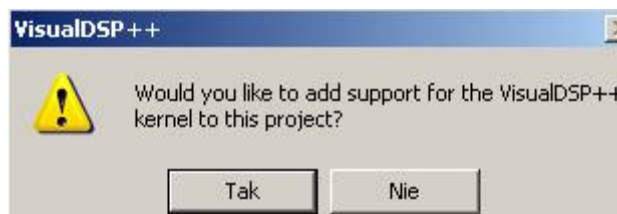


to samo tyczy się zakładki **Assemble**

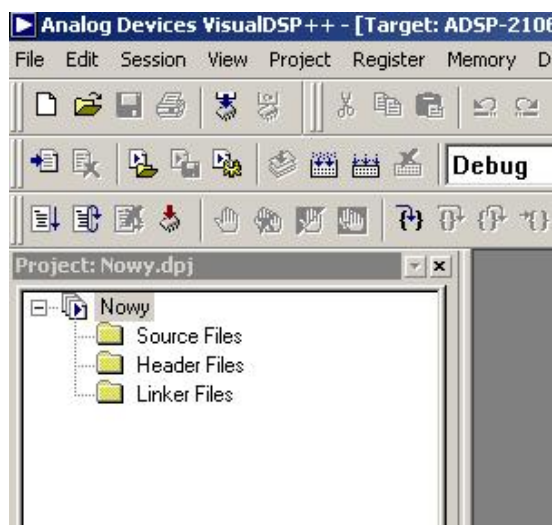


...a także zakładki **Link**

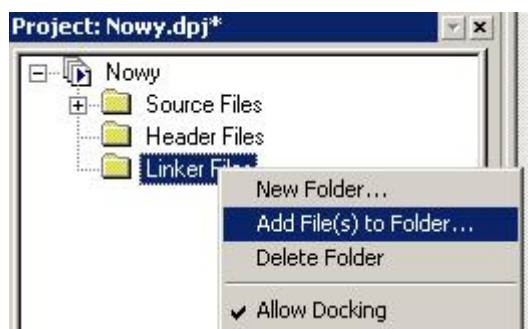
wciśnij **OK**



wybierz **Nie**



Efektom naszych działań jest utworzenie projektu o nazwie **Nowy**



Ale to jeszcze nie koniec! Aby projekt był gotowy do pracy musimy dodać do **Linker Files** pliki **\*.ldf** które dla procesora 21065L znajdziemy w katalogu **ProgramFiles/AnalogDevice/VisualDSP/21k/ldf**

Wybieramy plik o nazwie **ADSP-21065L\_C**

**Być może zastanawiasz się dlaczego właśnie takie a nie inne opcje zostały wybrane i cóż one oznaczają? Być może jesteś lekko zirytowany i rozczarowany.**

**Otóż naszym celem jest szybkie i sprawne rozpoczęcie pracy w nowym środowisku jakim jest **VisualDSP++**.**

**Wszelkie rozważania, dodatkowe opisy są tutaj zbędne lub inaczej; przyjdzie na nie czas później.**

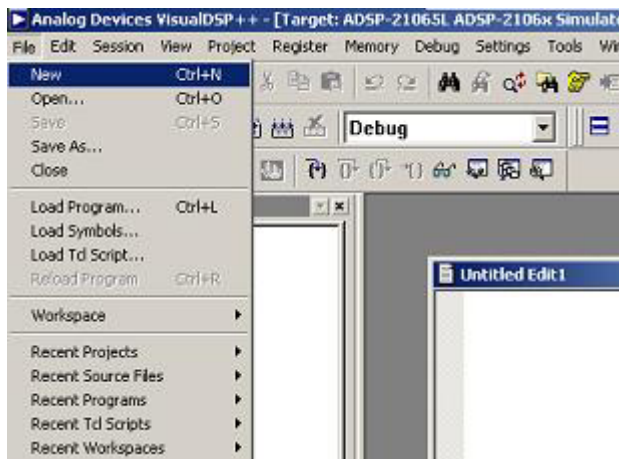
**Skupmy się na tym co jest najistotniejsze!**

**Mój opis nie jest żadnym podręcznikiem czy instrukcją, ale swego rodzaju przewodnikiem, którego celem jest tak poprowadzić czytelnika, aby samodzielnie mógł napisać oraz zasymulować choć najprostszy program.**

## ❑ Mój pierwszy program

Skoro mamy już otwarty nasz świeżo utworzony projekt „Nowy” to kolejnym krokiem jest umieszczenie w tym projekcie jakiegoś programu.

W ramach jednego projektu możemy realizować wiele programów. Projekt sam w sobie jest jedynie pewną bazą wyjściową jeśli chodzi o ustawienia symulatora (wybór procesora oraz innych w tej chwili nieistotnych rzeczy)

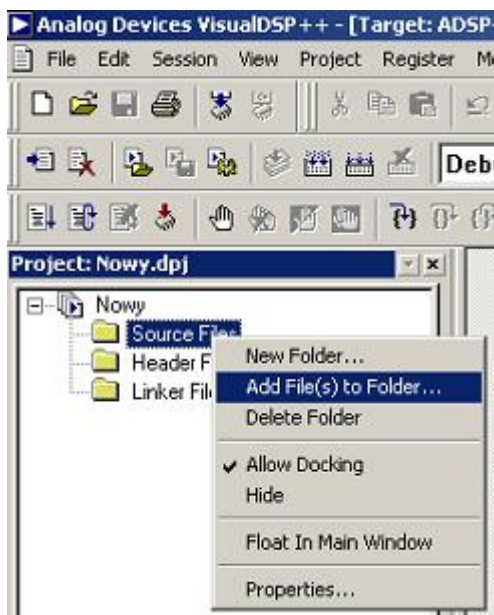


Z paska wybieramy **File/New**

Pojawi się nowe okienko **Untitled Edit 1**

Z paska wybieramy **File/Save As...** i zapisujemy plik jako **p1.c**

Nazwa może być dowolna ale musi mieć rozszerzenie **\*.c**



Teraz musimy nowo utworzony plik **p1.c** dodać do projektu.

Prawym klawiszem myszy klikamy na **Source File** i wybieramy **Add File(s) to Folder...**

Następnie wskazujemy na nasz **p1.c** i sprawa załatwiona.

Teraz możemy zacząć pisać nasz program posługując się naturalnie

językiem C.

Nie będę tutaj mówił jak pisać programy w C. Zakładam że posiadasz potrzebną wiedzę na ten temat.

Jeśli jednak nie jesteś zbyt biegły w programowaniu i napisanie jakiegokolwiek programu w C będzie dla Ciebie kłopotem to w tym właśnie miejscu odsyłam Cię na szybki kurs [www.kursc.prv.pl](http://www.kursc.prv.pl)

**W przeciwnym wypadku... lecimy dalej!**

```
// Program p1 do próbkowania sygnału sinus
#include <stdio.h>
#include <math.h>

#define PI 3.141592654
#define N 30
float dm d[N];

float dane_wejscowe(void);

main(void)
{
    dane_wejscowe();

    return(0);
}

float dane_wejscowe(void)
{
    int i;
    for(i = 0; i < N; i++)
        d[i]=sin(2*PI*i/N);

    return(0);
}
```

Tak wygląda kod dość prostego programu który próbkuje sygnał sinusoidalny a wartości próbek wpisuje do tablicy **d[N]**

```
// Program p1 do próbkowania sygnału sinus
#include <stdio.h>
#include <math.h>

#define PI 3.141592654
#define N 30
float dm d[N];

float dane_wejscowe(void);

main(void)
{
    dane_wejscowe();

    return(0);
}

float dane_wejscowe(void)
{
    int i;
    for(i = 0; i < N; i++)
        d[i]=sin(2*PI*i/N);

    return(0);
}
```

Skopiuj powyższy kod i wklej go do swojego projektu. Postaraj się dość sumiennie go przeanalizować.

Aby skompilować/zasymulować nasz **p1.c** należy z wybrać menu **Project/Rebuild All**

Gdy program zatrzymie się w takim punkcie zwanym **Breakpoint** wciśnij **F5**

Po chwili na dolnym pasku ukaże się napis **Halted** co będzie oznaczać (w tym przypadku) że program został już skompilowany.

**A więc program już działa! Ale nie widzimy w zasadzie żadnych efektów.**

**Co zrobić aby obejrzeć wyniki obliczeń?**

**Otóż należy....**

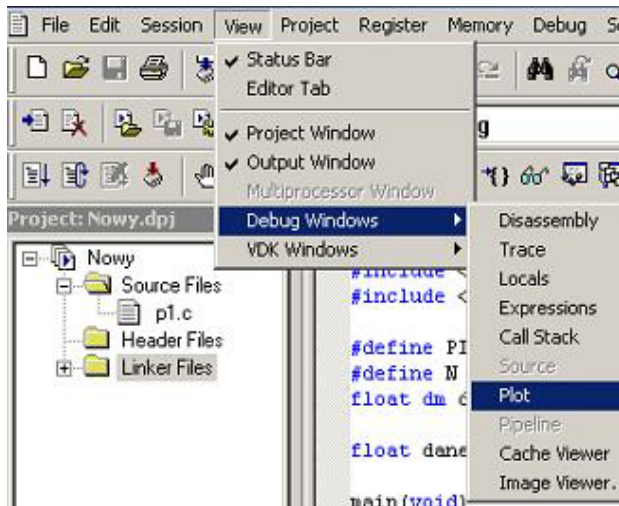
```
float dane_wejscowe(void);

main(void)
{
    dane_wejscowe();

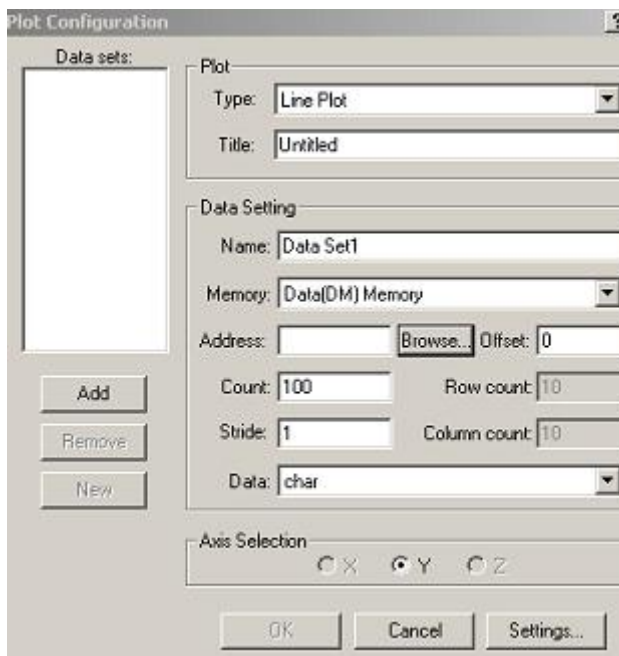
    return(0);
}

float dane_wejscowe(void)
{
```





Wybieramy:  
**View/DebugWindows/Plot/New...**



W tym okienku interesują nas takie opcje:

**Title:** Nazwa jaka nadamy naszemu wykresowi – a będzie to wykres przedstawiający wartości kolejnych zmiennych tablicy **d**

**Memory:** Ustawiamy w pozycji DM

**Address:** I w tym właśnie miejscu spójrz raz jeszcze na kod naszego programu! Zwróć szczególną uwagę na literki **pm** które znajdują się przed deklaracją zmiennej **d**, a wygląda to tak:

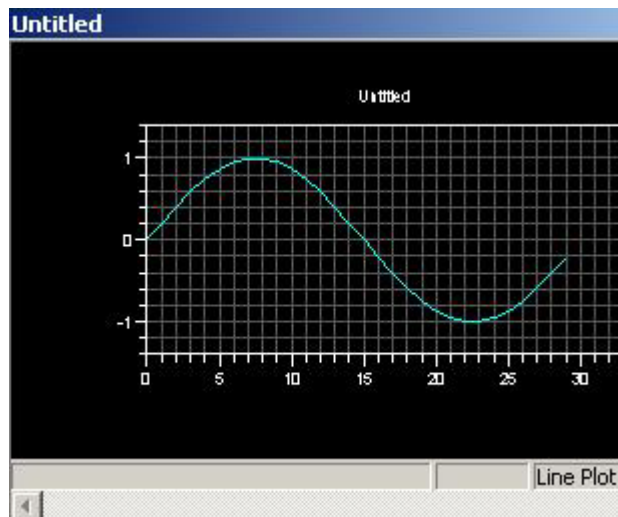
**float dm d[N];**

Dzięki temu naszą zmienną **d** możemy znaleźć wciskając **Browse...**

**dm – DataMemory**

**Count:** to liczba punktów wyświetlanych na osi **X**

**Data:** to typ zmiennej, w naszym przypadku jest to zmienna typu **float**



Na końcu należy kliknąć **Add** oraz **OK** i cieszyć się ładnym wykresem sinusoidy!

Funkcja **plot** posiada oczywiście wiele wiele opcji. Wystarczy w okienku **Plot Configuration** wcisnąć **Settings...**

Zalecam w tym miejscu zabawę dostępnymi opcjami.

Np. zmniejszenie szerokości marginesów wykresu czy też jego kolorystyki.

**Mam nadzieję że czas poświęcony na czytanie tego skromnego niezwykle wprowadzenia nie uważasz za stracony!**