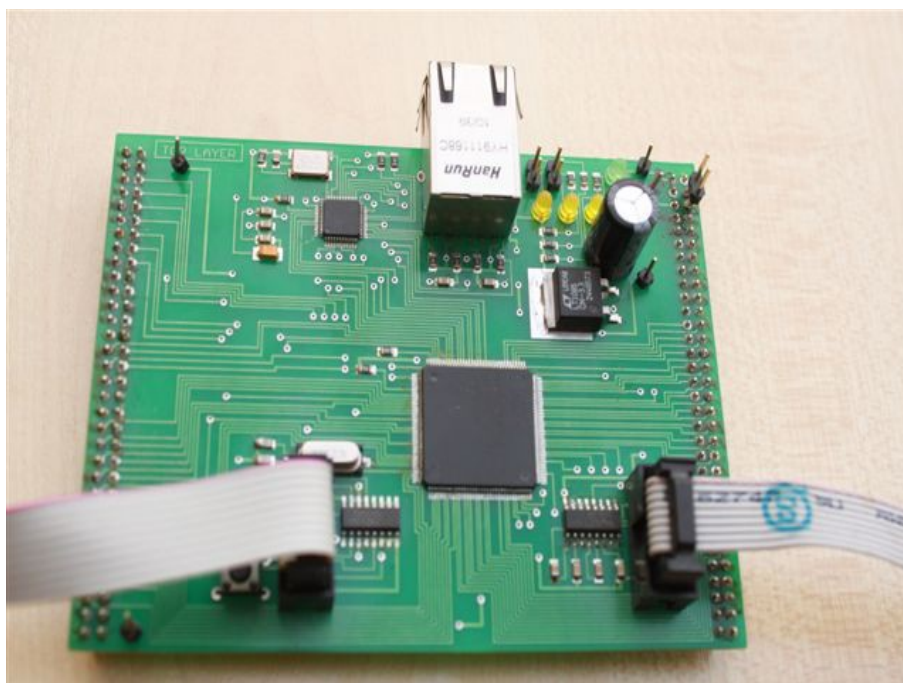


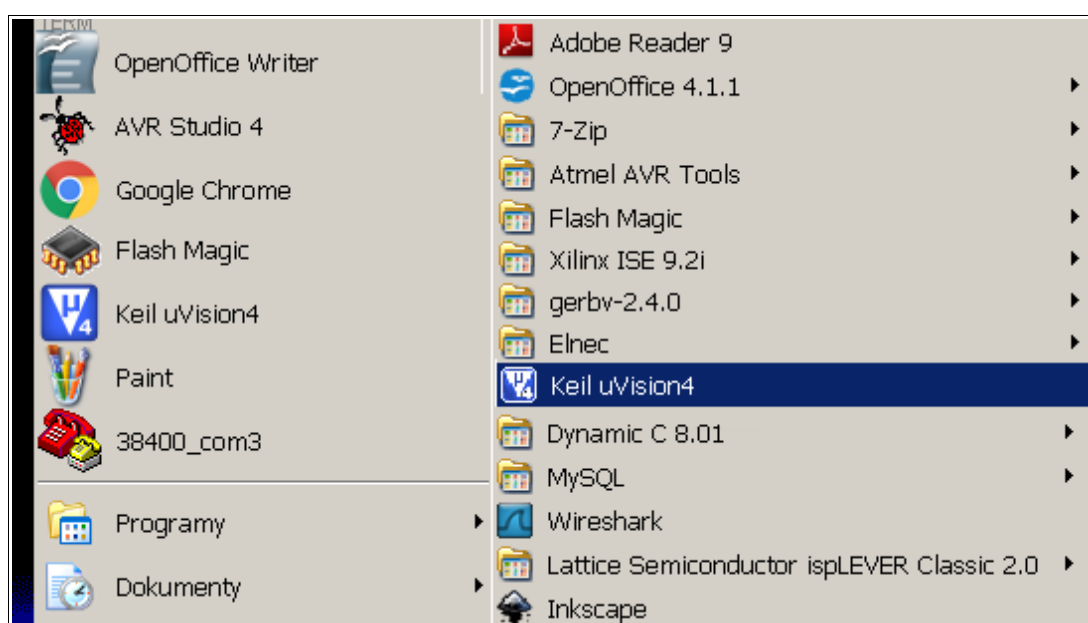
Andrzej Pawluczuk



***Filozofia sieci:
posługiwanie się środowiskiem
 μ Vision (Keil)***

*dla Elportal
Białystok, styczeń 2019*

Do tworzenia programów dla mikrokontrolerów ARM z rodziny LPC2000 może być zastosowane oprogramowanie narzędziowe opracowane przez firmę KEIL. Jest to oprogramowanie komercyjne, jednak może być stosowane do celów niekomercyjnych (posiada przy tym ograniczenie na wielkość generowanego kodu do 32kB, które w wielu przypadkach nie stanowi jakiegokolwiek przeszkody). Oprogramowanie można pobrać ze strony www.keil.com (ja posługuję się programem w nieco starszej wersji *Mdk474.exe*, chociaż są dostępne nowsze). Jego instalacja przebiega typowo i nie powinna sprawić jakiegokolwiek problemu. Program uruchamia się wybierając odpowiednią pozycję w menu (ilustracja 1).



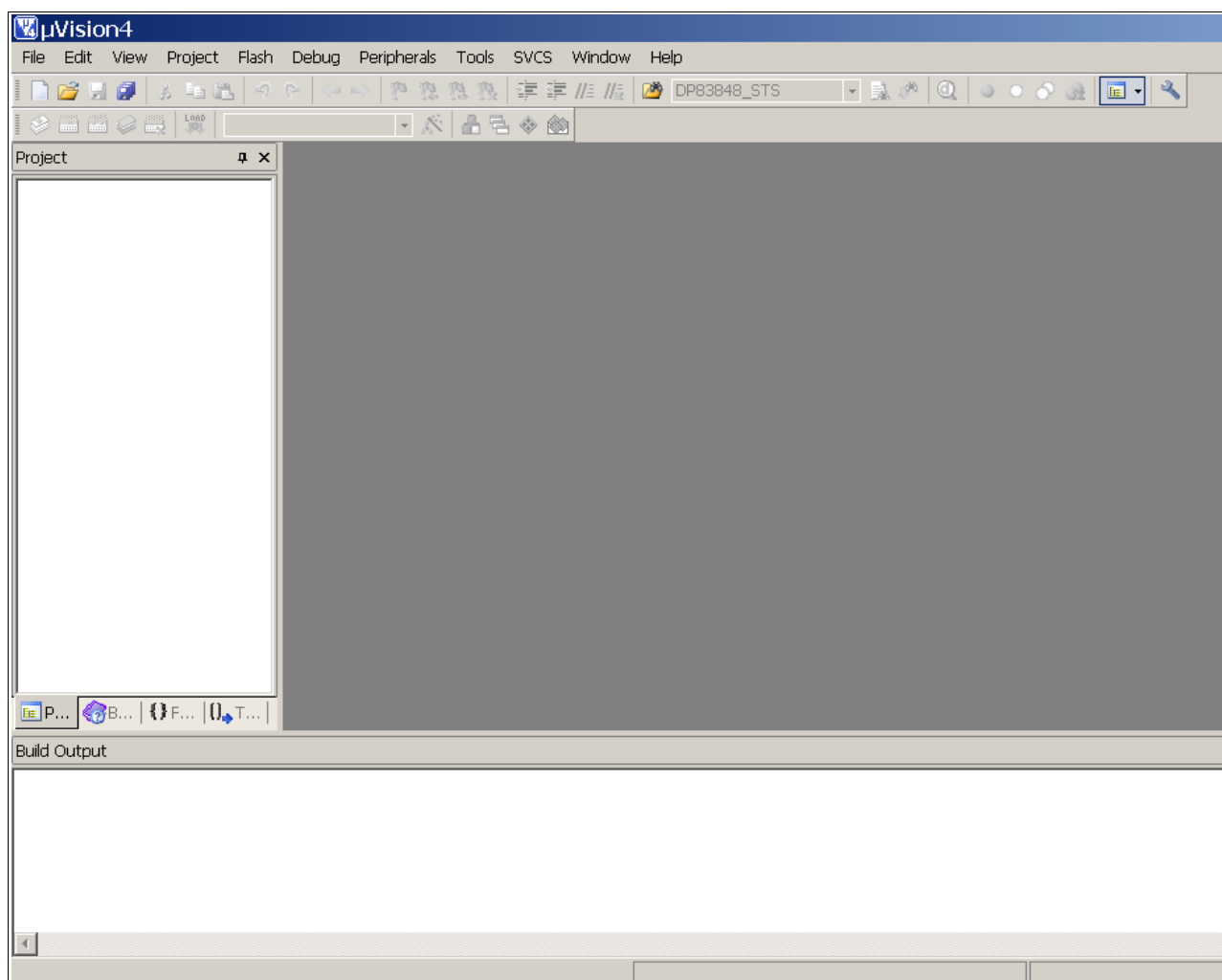
Ilustracja 1: Uruchomienie programu

Uruchomienie jest sygnalizowane „winietką” (ilustracja 2). Pełne uruchomienie pozwala na tworzenie nowego projektu lub kontynuację prac nad istniejącym. Program μ Vision automatycznie otwiera projekt, nad którym były prowadzone ostatnio prace (który nie został zamknięty).

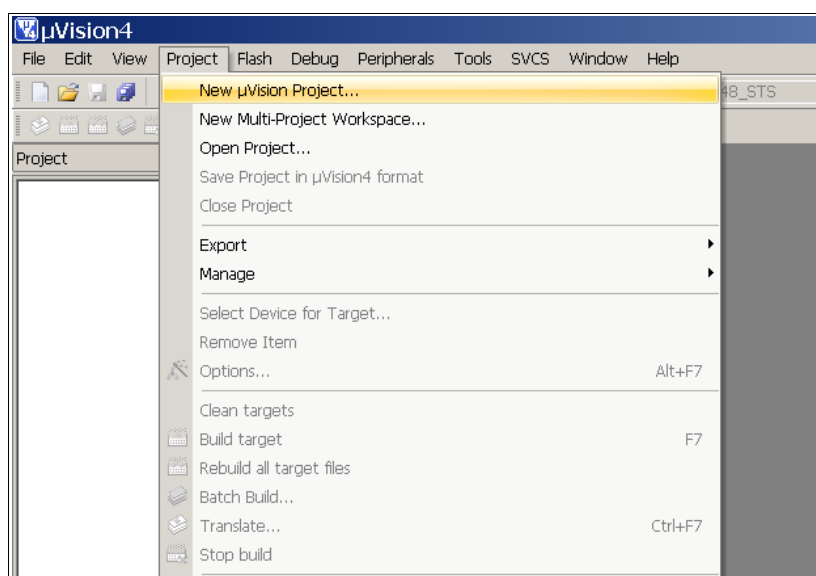


Ilustracja 2: Start oprogramowania

Środowisko gotowe do pracy (ilustracja 3).

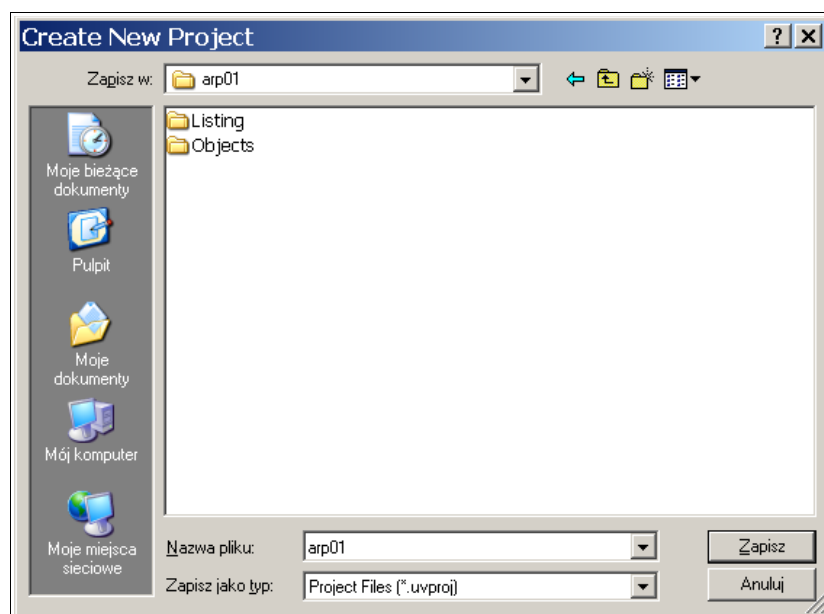


Ilustracja 3: μ Vision gotowy do działania



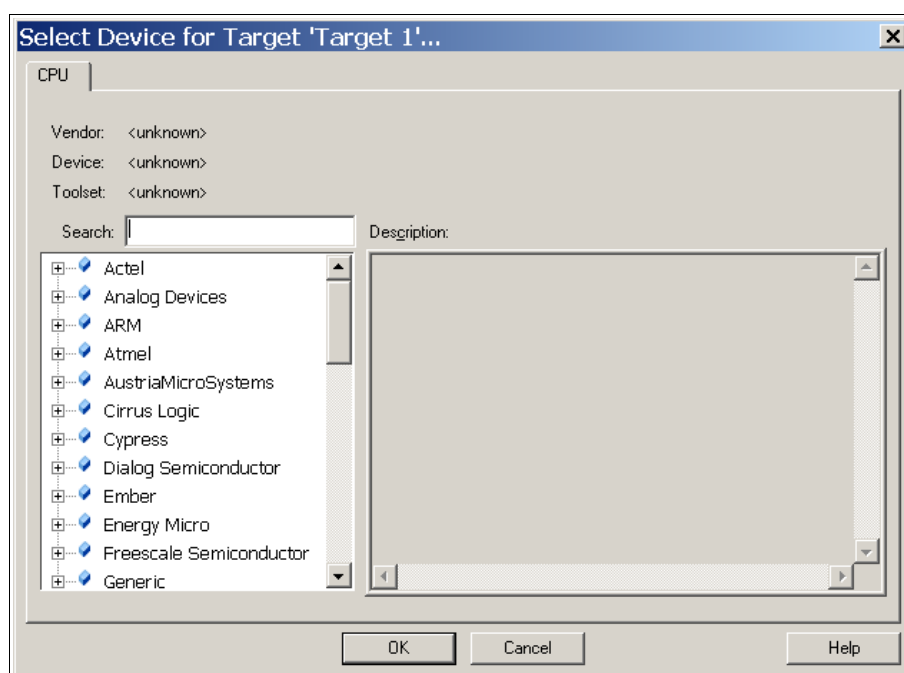
Ilustracja 4: Tworzenie nowego projektu

Utworzenie nowego projektu następuje po wybraniu opcji „Project → New μ Vision Project...” (ilustracja 4). Prowadzi to do otwarcia okienka dialogowego, które jest przeznaczone do wskazania lokalizacji projektu (położenia projektu w strukturze kartotek na dysku) oraz wprowadzenia nazwy projektu (ilustracja 5).



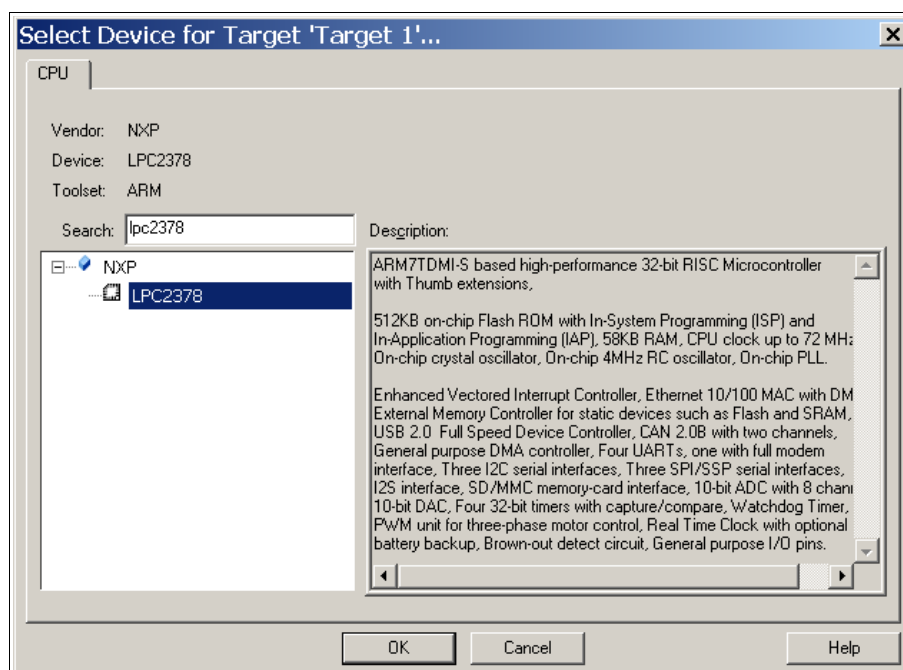
Ilustracja 5: Wprowadzenie nazwy projektu

Kolejnym krokiem jest określenie docelowego modelu mikrokontrolera, dla którego tworzone jest oprogramowanie (ilustracja 6).

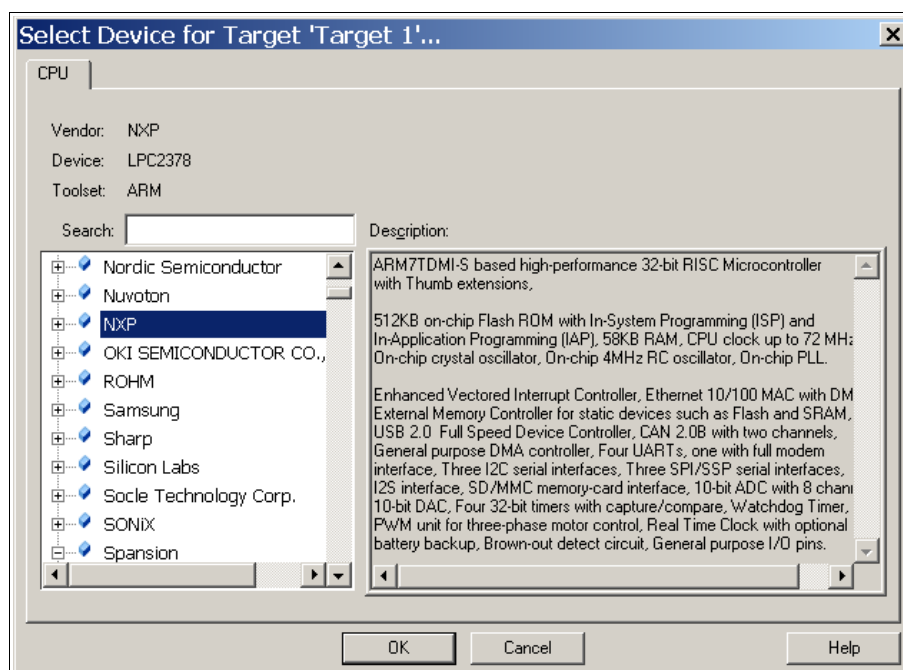


Ilustracja 6: Wybranie modelu mikrokontrolera

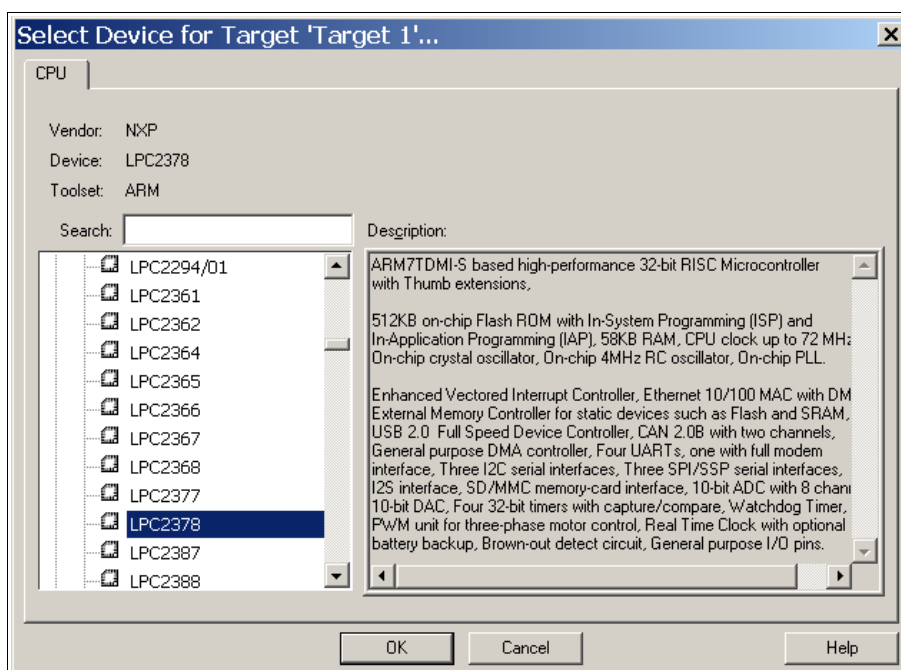
Można w okienku *Search* wpisać model i zaakceptować wybór klikając na „OK” (ilustracja 7) lub wybrać producenta (ilustracja 8), klikając na „+” wybrać model (ilustracja 9) i wybór zaakceptować klikając na „OK”.



Ilustracja 7: Wybór procka

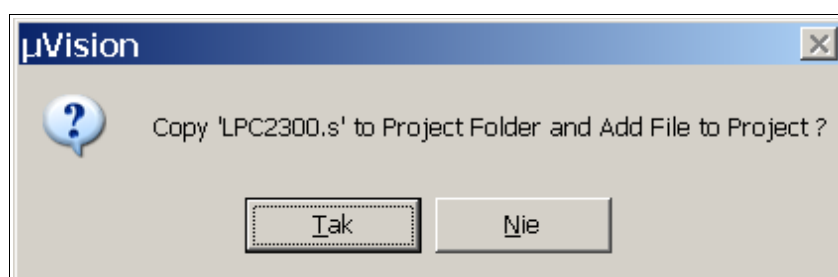


Ilustracja 8: Wybór producenta

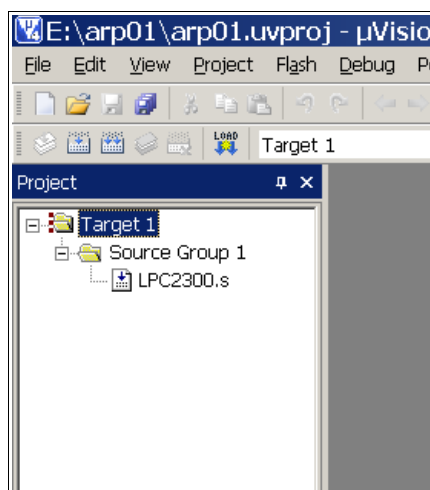


Ilustracja 9: Wybór modelu mikrokontrolera

Prowadzi to do dodania do projektu pierwszego pliku źródłowego, jakim jest startup (część rozruchowa dla danego mikrokontrolera). Plik pochodzi z „zapasów” środowiska μ Vision i kompilator pyta, czy mamy ochotę taki plik dołączyć (istnieje możliwość utworzenia własnego, jakiegoś niestandardowego pliku rozruchowego, ale to już „inna półka”) jak pokazuje ilustracja 10. Należy odpowiedzieć TAK.



Ilustracja 10: Pytanie

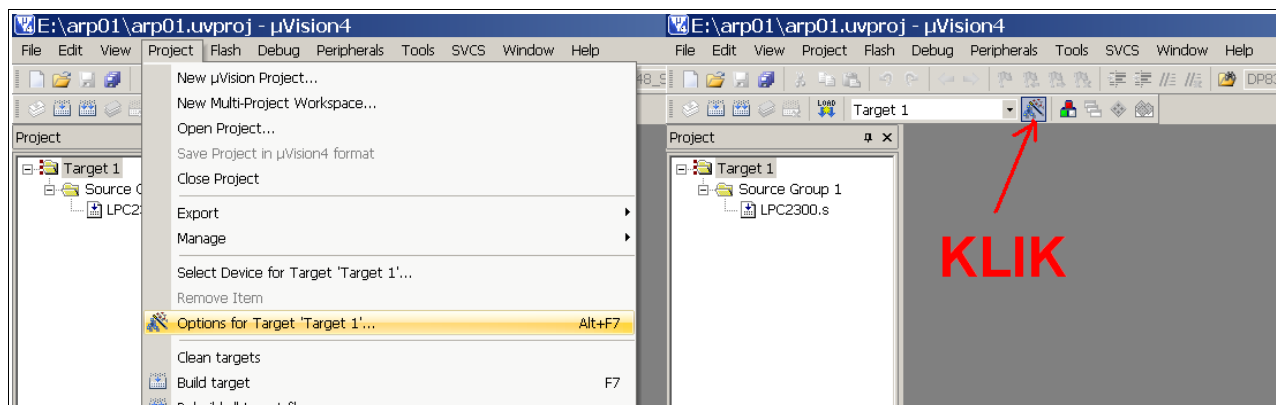


Ilustracja 11: "Pusty" projekt

Zostaje utworzony projekt zawierający już jeden plik (ilustracja 11). Plik o nazwie *LPC2300.s* jest częścią rozruchową dla tworzonego programu i zawiera już skonfigurowanie układu PLL (powielenie częstotliwości taktującej mikrokontroler), toteż nie należy tego robić w tworzonym programie.

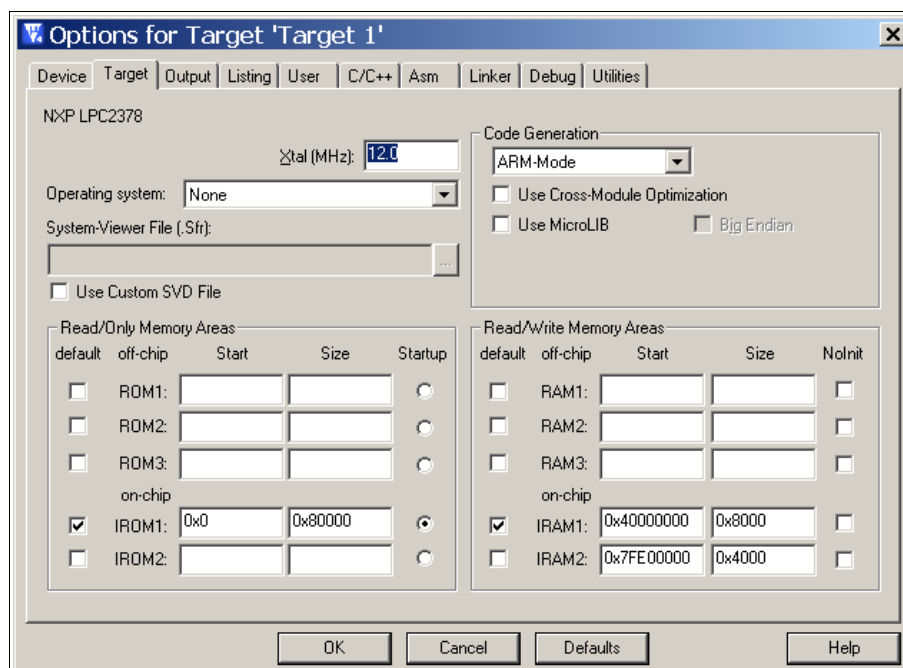
Zanim zostanie wygenerowany pierwszy kod programu należy zmodyfikować odpowiednie szczegóły związane z projektem. Dojście do szczegółów może być wykonane na kilka sposobów: można wyklikać odpowiednią opcję („Project →

Options for target...”), można użyć gotowego skrótu lub użyć kombinacji klawiszowej Alt+F7 (ilustracja 12).



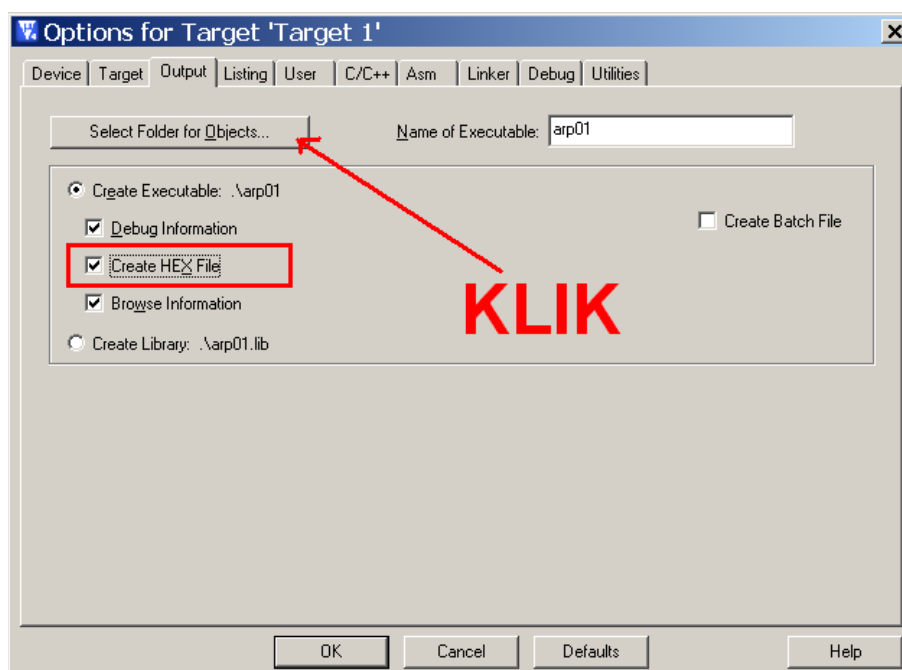
Ilustracja 12: Modyfikacja szczegółów projektu

Otwiera to okienko dotyczące szczegółów projektu. W zakładce „Target”, (ilustracja 13) właściwie nic nie należy zmieniać (ewentualnie częstotliwość rezonatora kwarcowego [Xtal]).



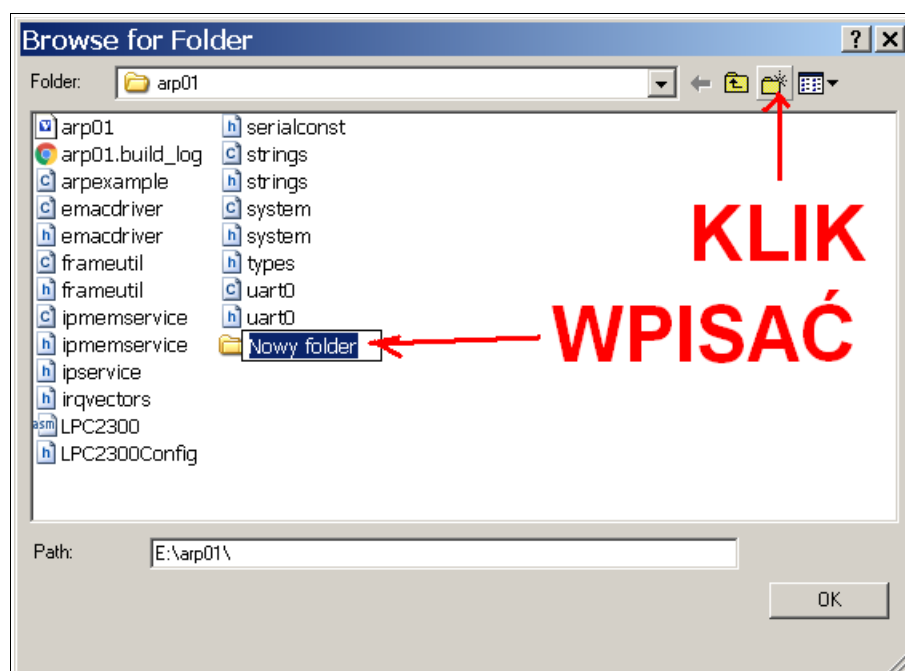
Ilustracja 13: Zakładka „Target”

W zakładce „Output” należy zaznaczyć opcję „Create HEX File” i warto klikając na „Select folder...” wskazać miejsce (kartotekę) na generowane pliki binarne, pozwoli to zapanować nad generowanym „śmietnikiem” (ilustracja 14).



Ilustracja 14: Zakładka „Output”

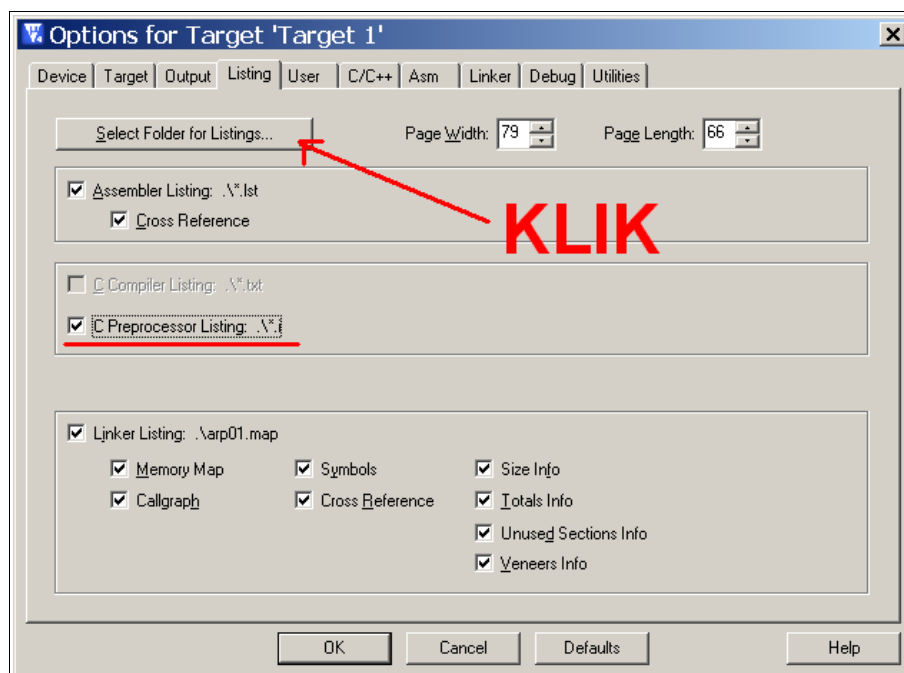
Daje to możliwość wskazania lub utworzenia nowej kartoteki przeznaczonej na binaria (ilustracja 15). Proponuję utworzyć nową o nazwie *OBJECT*. Po wpisaniu zrobić dwuklik by przejść do utworzonej i tam kliknąć na „OK” (wybrana jest ta kartoteka, w której aktualnie się znajdujemy).



Ilustracja 15: Kartoteka na object'y

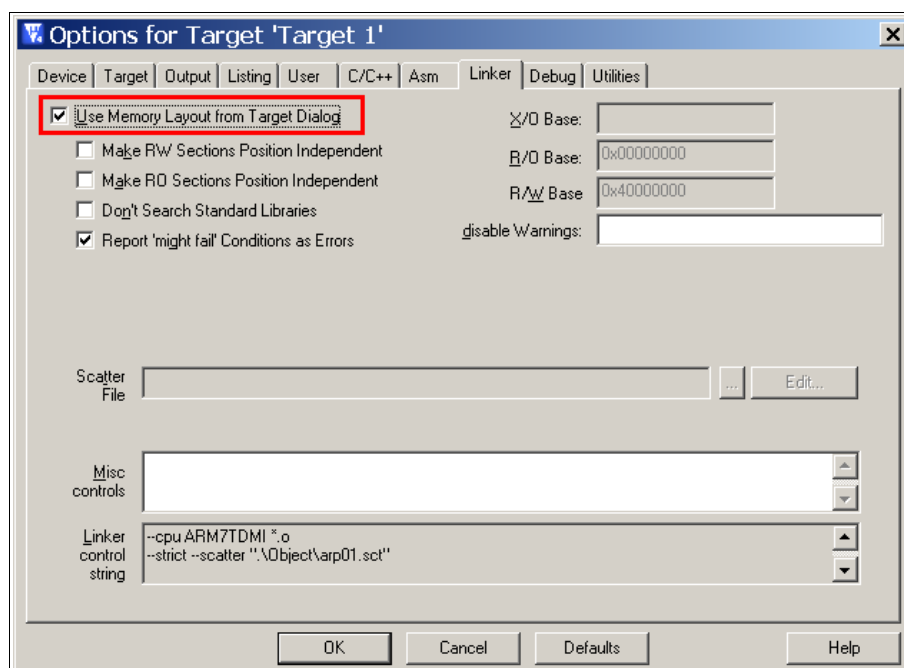
W zakładce „Listing” również warto wskazać kartotekę na listingi generowane przez kompilator. Klikając na „Select Folder for Listings” (ilustracja 16) powstaje identyczna jak w zakładce „Output”, możliwość utworzenia lub wskazania na

kartotekę. Proponuję utworzyć kartotekę o nazwie *LISTING*. Czasami warto zostawić pliki po przejściu preprocesora (to są „prawdziwe” pliki podlegające kompilacji, wszelkie zapisy typu `#define`, `#include` są poleceniami dla preprocesora, który generuje postać źródłową do kompilacji, typowo te pliki są usuwane, więc czasami jak trzeba przejrzeć „prawdziwy” tekst programu, to konieczne jest postawienie odpowiedniego „ptaszka”), ilustracja 16.



Ilustracja 16: Zakładka „Listing”

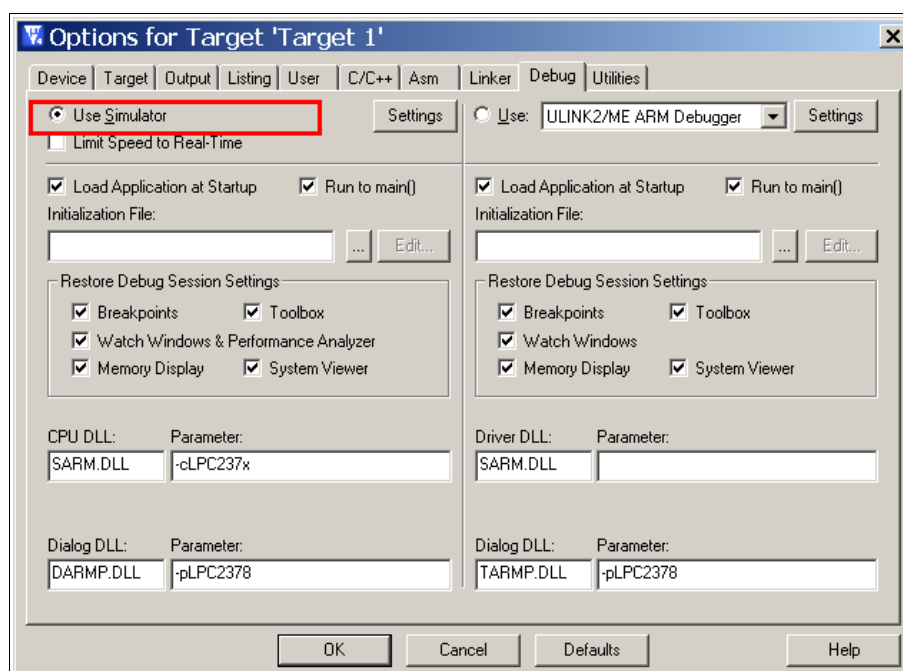
Przechodzimy do zakładki „Linker” (ilustracja 17).



Ilustracja 17: Zakładka „Linker”

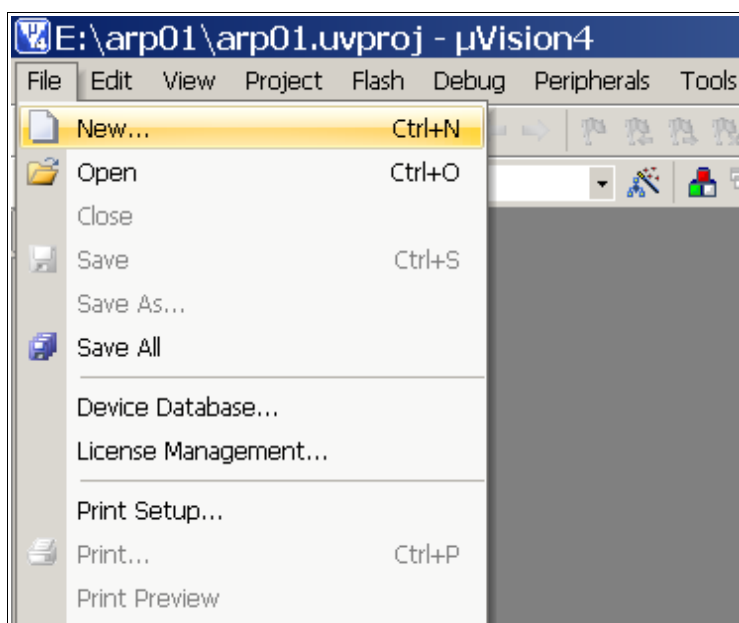
Tam należy zaznaczyć „ptaszka” w „Use Memory Layout from Target Dialog”. Brak zaznaczonej tej opcji spowoduje wygenerowanie błędnego kodu programu.

W zakładce „Debug” można zaznaczyć opcję „Use Simulator”, pozwoli to na symulację działania skompilowanego programu (ilustracja 18).

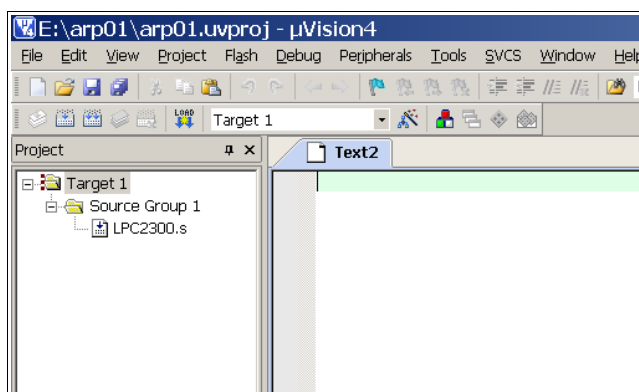


Ilustracja 18: Zakładka „Debug”

Po tych czynnościach mamy skonfigurowany projekt. Można przystąpić to napisania programu. Utworzenie pliku „od zera” polega na wybraniu odpowiedniej opcji: „File → New” (ilustracja 19).



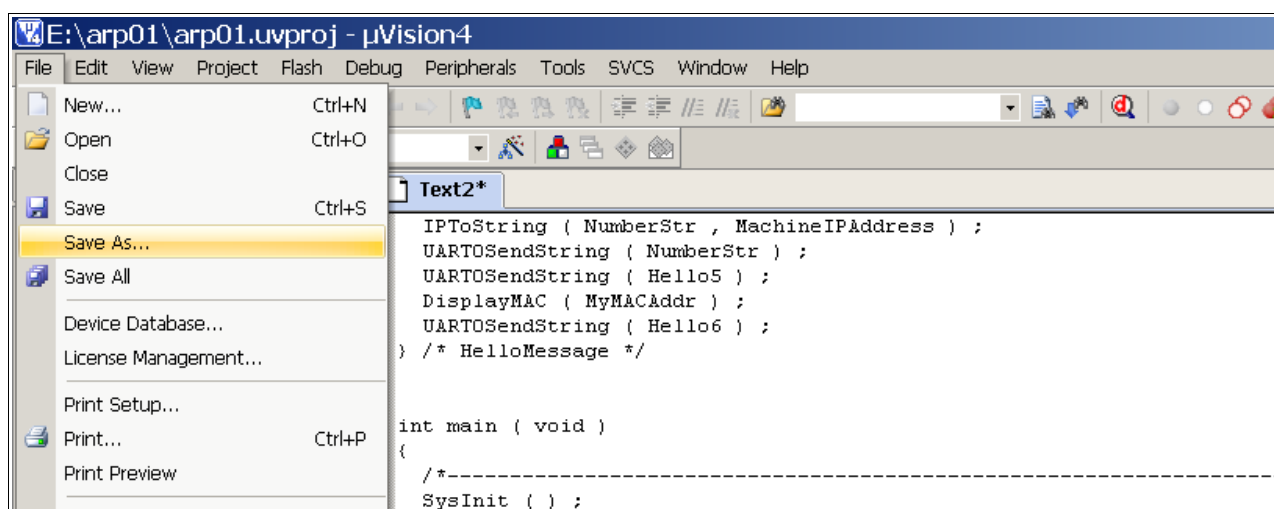
Ilustracja 19: Utworzenie nowego pliku



Ilustracja 20: "Pusty" plik w edytorze

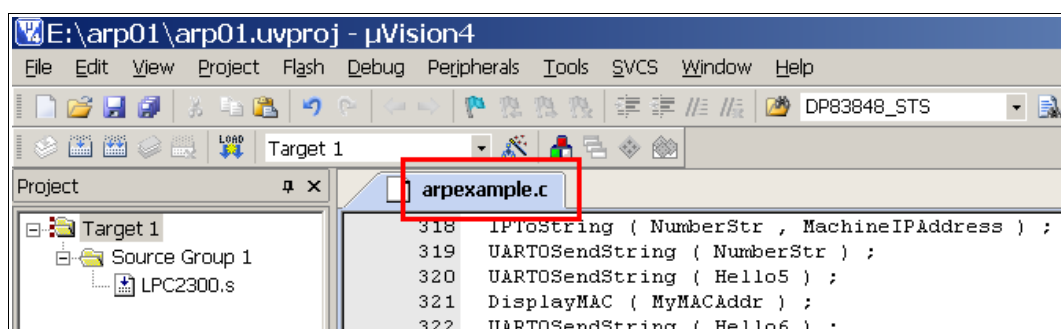
Daje to otwarcie „pustego pliku” w edytorze (ilustracja 20). Teraz pozostało jedynie wklepać tekst programu. Po zakończeniu wpisywania plik należy zapisać na dysku i jednocześnie nadać mu nazwę. Wybierając opcję „File → Save as” (ilustracja 21), powstaje taka możliwość. Należy pamiętać, by zapisywanemu plikowi nadać rozszerzenie typowe dla plików źródłowych zawierających tekst

programu w języku C (*.C).



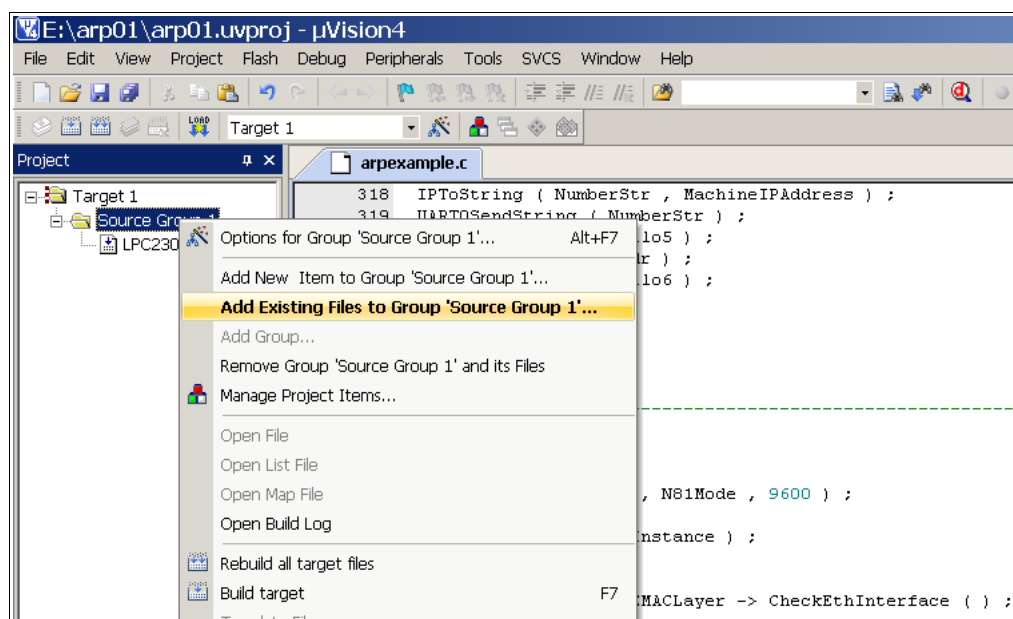
Ilustracja 21: Nazwanie pliku

Po tej operacji mamy plik nazwany (ilustracja 22). Nie znaczy to, że ten plik jest dołączony do projektu.



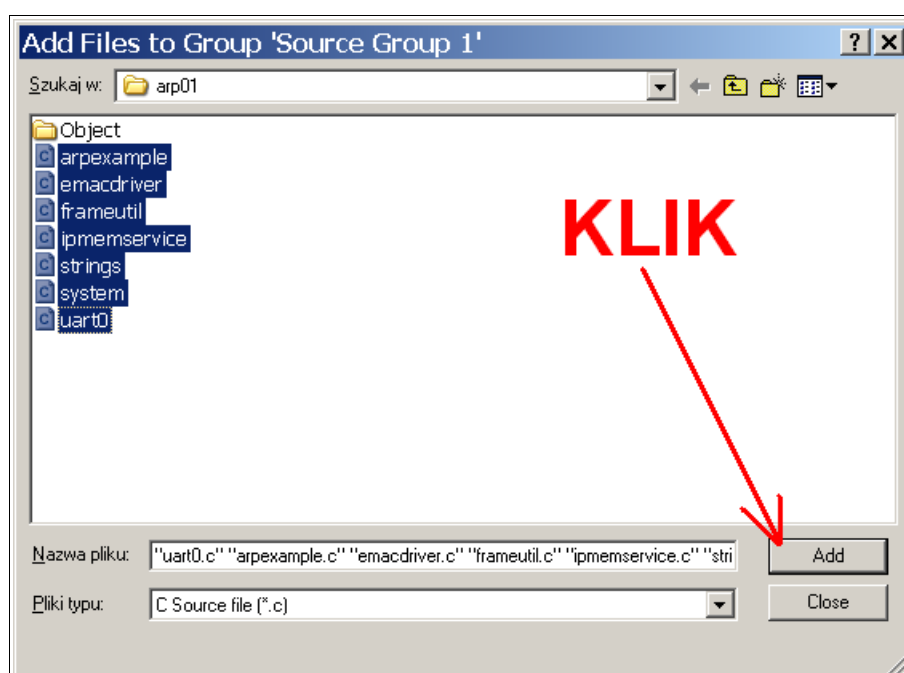
Ilustracja 22: Nazwany plik

By dołączyć jakikolwiek plik źródłowy do projektu, należy kliknąć prawym klawiszem myszki na „Source Group” i z rozwiniętego menu wybrać opcję „Add Existing Files to Group” (ilustracja 23). Powoduje to otwarcie odpowiedniego okienka dialogowego (ilustracja 24).



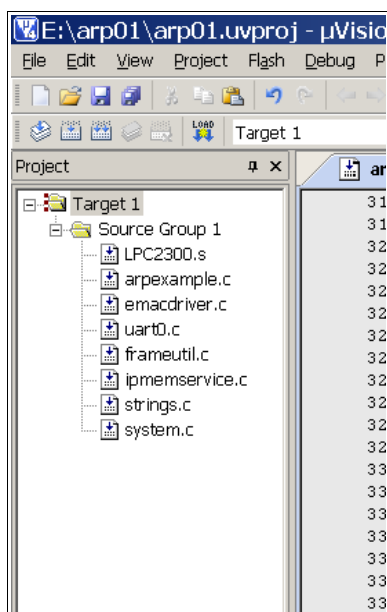
Ilustracja 23: Dodawanie plików do projektu

Klikając myszką na nazwy plików można zaznaczać, które mają być dołączone (zaznaczanie kolejnego wymaga trzymania klawisza Ctrl). Po wybraniu niezbędnych plików należy kliknąć na „Add” (można wybierać pliki pojedynczo i za każdym razem kliknąć na „Add”). Po zakończeniu plików, okienko zamyka się po kliknięciu na „Close”.



Ilustracja 24: Wybór plików

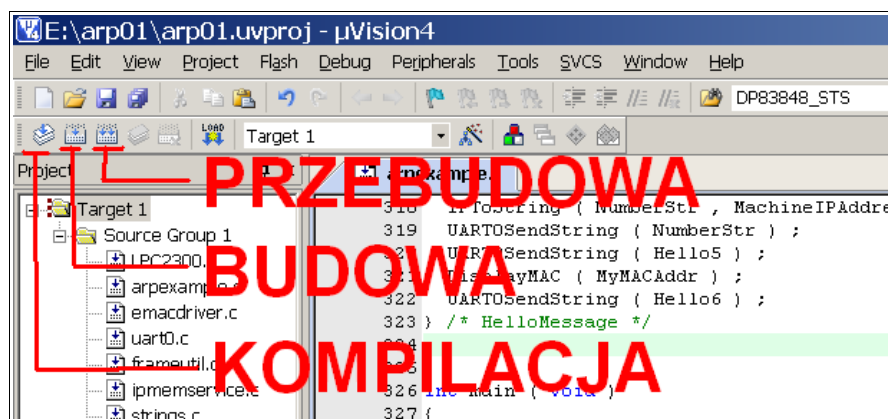
Do projektu są dołączone wszystkie niezbędne pliki źródłowe (ilustracja 25).



Ilustracja 25: Pliki projektu

Nadszedł czas na kompilację. Mamy kilka wariantów (ilustracja 26):

- tylko kompilacja – kompilacja pliku źródłowego znajdującego się w aktywnym okienku,
- budowa – kompilacja i linkowanie wszystkich plików, które zostały zmienione od czasu ostatniej kompilacji,
- przebudowa – kompilacja i linkowanie wszystkich plików wchodzących w skład projektu.



Ilustracja 26: Możliwości kompilacji

Po wykonaniu „budowy” lub „przebudowy”, jeżeli nie zaistniały żadne błędy, zostaje wygenerowany plik jako wsad do pamięci mikrokontrolera (plik w formacie Intel-hex). Będzie on znajdował się w kartotece przeznaczony na pliki wynikowe (OBJECT), toteż tam należy go poszukiwać w operacji programowania pamięci FLASH przy zastosowaniu programu FLASHMAGIC.

Zalecana dodatkowa lektura:

1. *„Posługiwanie się programem FLASHMAGIC”*