

AVR-GCC: kompilator C mikrokontrolerów AVR, część 6



Do używanych w projekcie plików nagłówkowych *.h odwołujemy się w dwojaki sposób: `#include <file.h>` albo `#include „file.h”`. Różnica leży tylko w sposobie wyszukiwania przez kompilator na dysku pliku o podanej nazwie. Przy odwołaniu `<>` w pierwszej kolejności sprawdzane są własne, systemowe zasoby plików `avr-gcc` (a więc foldery `avr\include` i `lib\gcc\avr\wersja\include`), do których nie musimy podawać ścieżki. Dołączamy więc takim zapisem wszystkie potrzebne w projekcie pliki `avr-libc`. Konieczne jest jednak zaznaczenie wejścia do ewentualnych subfolderów a więc np. `#include <avr/io.h>`. Użycie uniksowego slasha `/` zamiast windowsowego backslasha `\` nie jest błędem: kompilator interpretuje go poprawnie natomiast znacznie ułatwione jest przeniesienie projektu do środowiska uniksowego (jak Linuks).

Forma „” uruchamia wyszukiwanie pliku od bieżącego folderu projektu oraz dodatkowych podanych kompilatorowi lokalizacji. W ten sposób powołujemy się więc na własne pomocnicze pliki nagłówkowe projektu (jak `dane.h` w naszym przykładzie). Dodatkowe ścieżki przeszukiwania wprowadzamy przy pomocy opcji `-Iścieżka`. Środowisko `AvrSide` wspiera jedną pomocniczą lokalizację, którą wpisujemy w oknie edycyjnym dialogu *Konfiguracja projektu* > *Ścieżki*. Zazwyczaj będzie to folder `[AvrSide\Myinc]` przewidziany na ogólne własne pliki nagłówkowe z ulubionymi typami, definicjami, pomocniczymi makrami używanymi w wielu projektach. Zauważmy jednak, że narzucenie konkretnej pełnej ścieżki dostępu może sprawiać kłopoty przy przenoszeniu projektu na inną maszynę z inaczej zainstalowanym `AvrSide` – po prostu takiej ścieżki może nie być co spowoduje błędy. Dla takich przenośnych zastosowań przewidziałem dodatkowe subfoldery foldera projektu `[lib]` oraz `[inc]`. Posługują się one lokalizacją względną nie utrudniającą przenoszenia. Utworzenie tych subfolderów jest dokonywane samoczynnie przy zapisaniu projektu z ustawioną opcją *Konfiguracja projektu* > *Ścieżki*

W tej części kursu przedstawiamy zasady korzystania z plików nagłówkowych, które są jednym z najbardziej podstawowych narzędzi ułatwiających pracę z kompilatorem AVR-GCC.

> *Używaj lokalnych podkatalogów lib/inc*. Automatycznie jest także przy wywołaniu kompilatora dodawana ścieżka do subfoldera `[inc]`.

Z lokalizacją własnych plików nagłówkowych związana jest jeszcze jedna opcja `AvrSide`: *Konfiguracja projektu* > *AvrSide* > *Szukaj deklaracji w folderach inc*. Jest ona połączona z mechanizmem podpowiedzi deklaracji symbolu (F1). Przeszukanie plików w poszukiwaniu deklaracji odbywa się domyślnie wewnątrz foldera projektu. Zaznaczenie opcji powoduje również przeszukiwanie plików w dodatkowych lokalizacjach – jest to bardziej uniwersalne ale może spowalniać pracę `AvrSide` w przypadku nagromadzenia dużej liczby plików.

Następne ułatwienie w pracy z plikami nagłówkowymi dotyczy zależności. Mianem tym określamy w tym przypadku ustalenie, w których plikach źródłowych *.c oraz *.s jest używany dany plik nagłówkowy. Zmiana w takim pliku nagłówkowym pociąga za sobą oczywiście zmiany w tych źródłach co wymaga ich ponownego przekompilowania. Jednak bez samoczynnego wsparcia musielibyśmy albo zawsze pamiętać, których plików mogą dotyczyć zmiany albo zawsze po korekcie dowolnego nagłówka ponownie kompilować cały projekt (*Build*), co przy rozbudowanych programach może znacznie wydłużać całą operację. Obsługa zależności w `AvrSide` jest dosyć uproszczona: polega na przeszukaniu pierwszych kilkunastu wierszy kodu każdego źródła .c i .s i utworzeniu tabeli zależności na podstawie znalezionych dyrektyw dołączenia (`#include`) wszelkich otwartych w projekcie (obecnych na zakładkach edytora) plików nagłówkowych *.h. Tabela ta jest później sprawdzana przy zapisywaniu zmian w pliku nagłówkowym – na tej podstawie są oznaczane jako zmienione (podkreślenie nazwy pliku na zakładce edytora) i włączane do polecenia *Make* odpowiednie moduły źródłowe.

Dla przyspieszenia działania środo-

wiska aktualizacja tabeli zależności nie odbywa się samoczynnie; po wprowadzeniu w projekcie korekt takich jak dodanie lub usunięcie pliku czy też dopisanie (usunięcie) dołączenia nagłówka w kodzie źródłowym, należy użyć polecenia menu *Projekt* > *Aktualizuj zależności*. Nie musimy jednak tego każdorazowo robić przy ładowaniu projektu – w tym przypadku utworzenie tabeli jest automatyczne. (Powyższe polecenie jest także przydatne dla zresetowania tabeli w razie wystąpienia błędu, którego niestety nie udało mi się dotychczas zlokalizować, a który powoduje ciągle oznaczenie plików źródłowych jako zmienione pomimo ich przekompilowania).

Należy też pamiętać, że powyższa obsługa jest jednopoziomowa, nie wspiera zagnieżdżonych dołączeń (w nagłówku `1.h` dołączamy `2.h` a z kolei `1.h` jest dołączony do `3.c` – nasza uproszczona obsługa wykryje zmianę w `1.h` ale w `2.h` już nie, konieczne jest zastosowanie komendy *Build*). Jak widać nie może się więc ona równać z rozbudowanymi narzędziami kontroli zależności używanymi w plikach *makefile*. Jednak dość dobrze zdaje egzamin w przeciętnej wielkości amatorskich projektach, dla których przede wszystkim `AvrSide` powstało.

Często zbiór funkcji obsługujących konkretne zadanie (np. obsługę urządzenia peryferyjnego) lokujemy w oddzielnym module źródłowym `xx.c` a dostęp do niego realizujemy poprzez skojarzony plik nagłówkowy o takiej samej nazwie `xx.h`. `AvrSide` oferuje dodatkowe skróty klawiszowe (**SHIFT+F6** albo **ALT+UP**) do szybkiego przełączania pomiędzy tak nazwaną parą plików.

Jerzy Szczesiul, EP
jerzy.szczesiul@ep.com.pl

UWAGA!
Środowisko IDE dla AVR-GCC opracowane przez autora artykułu można pobrać ze strony <http://avrside.ep.com.pl>.