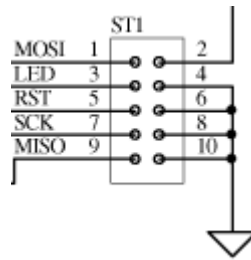


Jak programować AVRa w systemie, czyli bez konieczności wyciągania mikrokontrolera z układu ?

- Producent AVRów firma Atmel zaproponowała standard złącza 10-pinowego, jak na rysunku poniżej

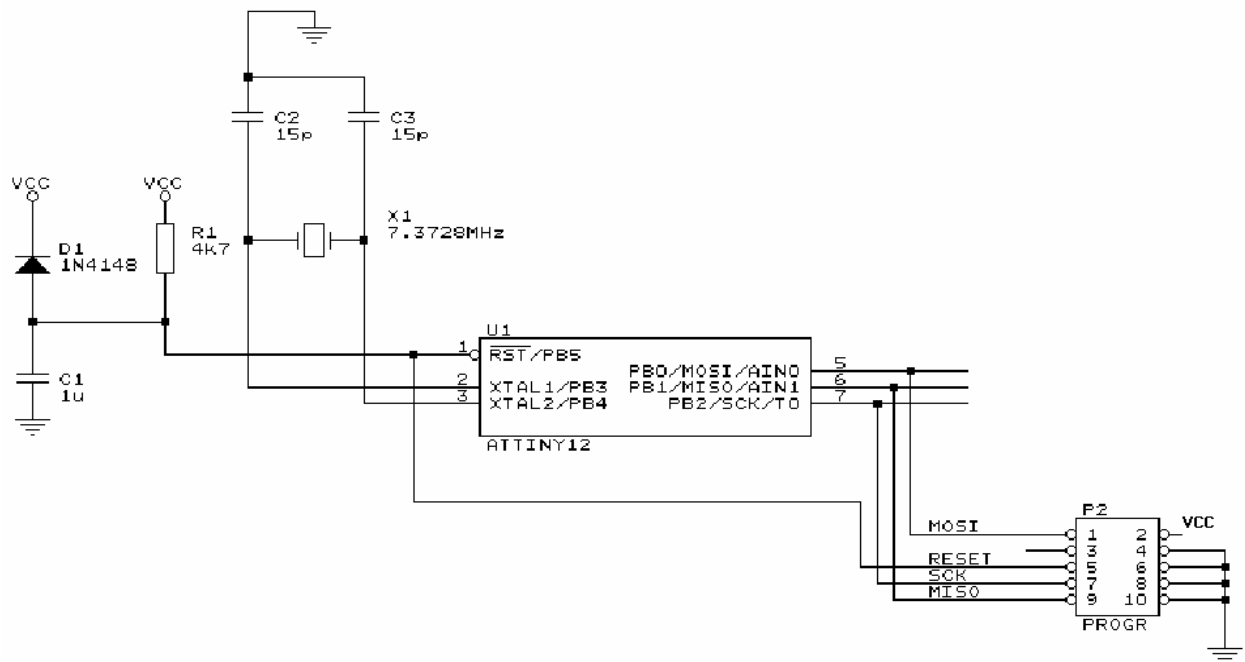


pin 2 = VCC
pin 4,6,8,10 = GND

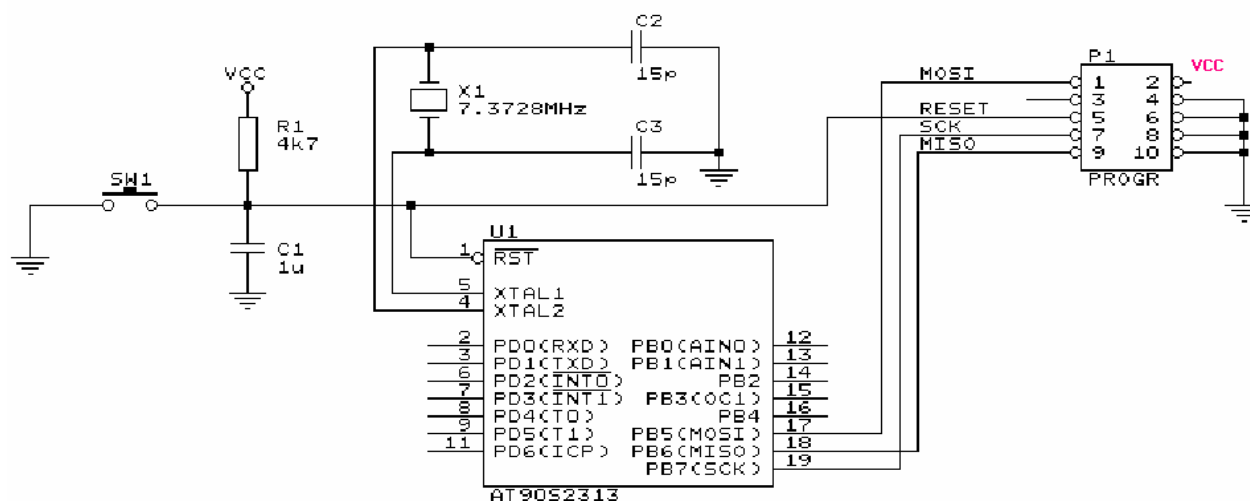
Gdzie **MOSI**, **MISO**, **SCLK** i **RESET** to sygnały służące do programowania, **GND** to masa układu, a **VCC** służy do zasilania programatora z tego samego źródła co programowany mikrokontroler tak, aby programator poprawnie współpracował pod względem poziomu napięć sygnałów logicznych. Wyjście **LED** służy do opcjonalnego podłączenia diody LED, która pokazuje stan pracy programatora. Dioda może być podłączana bezpośrednio między pin **LED** a **VCC**.

Układ należy więc wyposażyć w złącze 10-stykowe z pinami odpowiednio doprowadzonymi do zasilania i sygnałów programujących AVR'a.

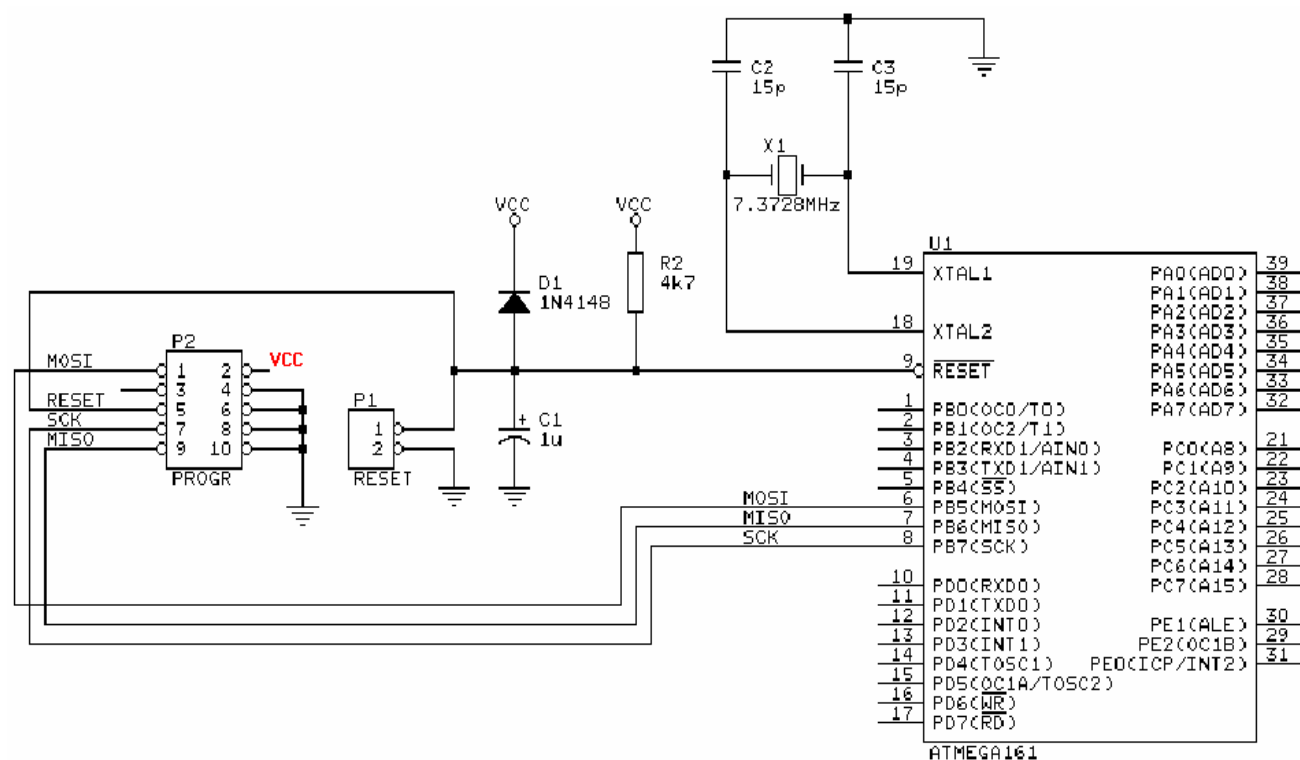
Przykłady podłączania programatora do mikrokontrolera AVR.



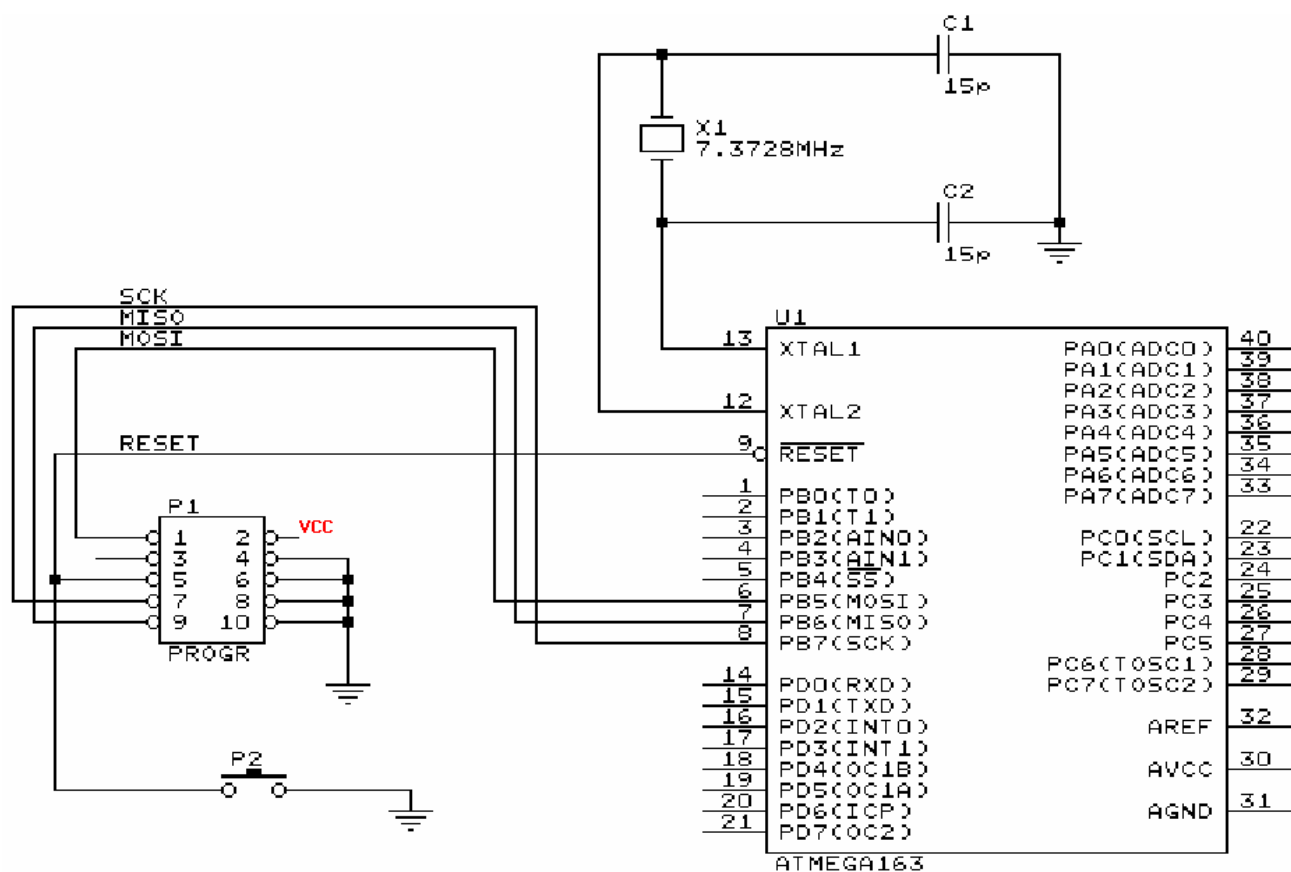
Przykład przyłącza do programatora dla mikrokontrolera Attiny12.



Przykład przyłącza do programatora dla mikrokontrolera AT90S2313.

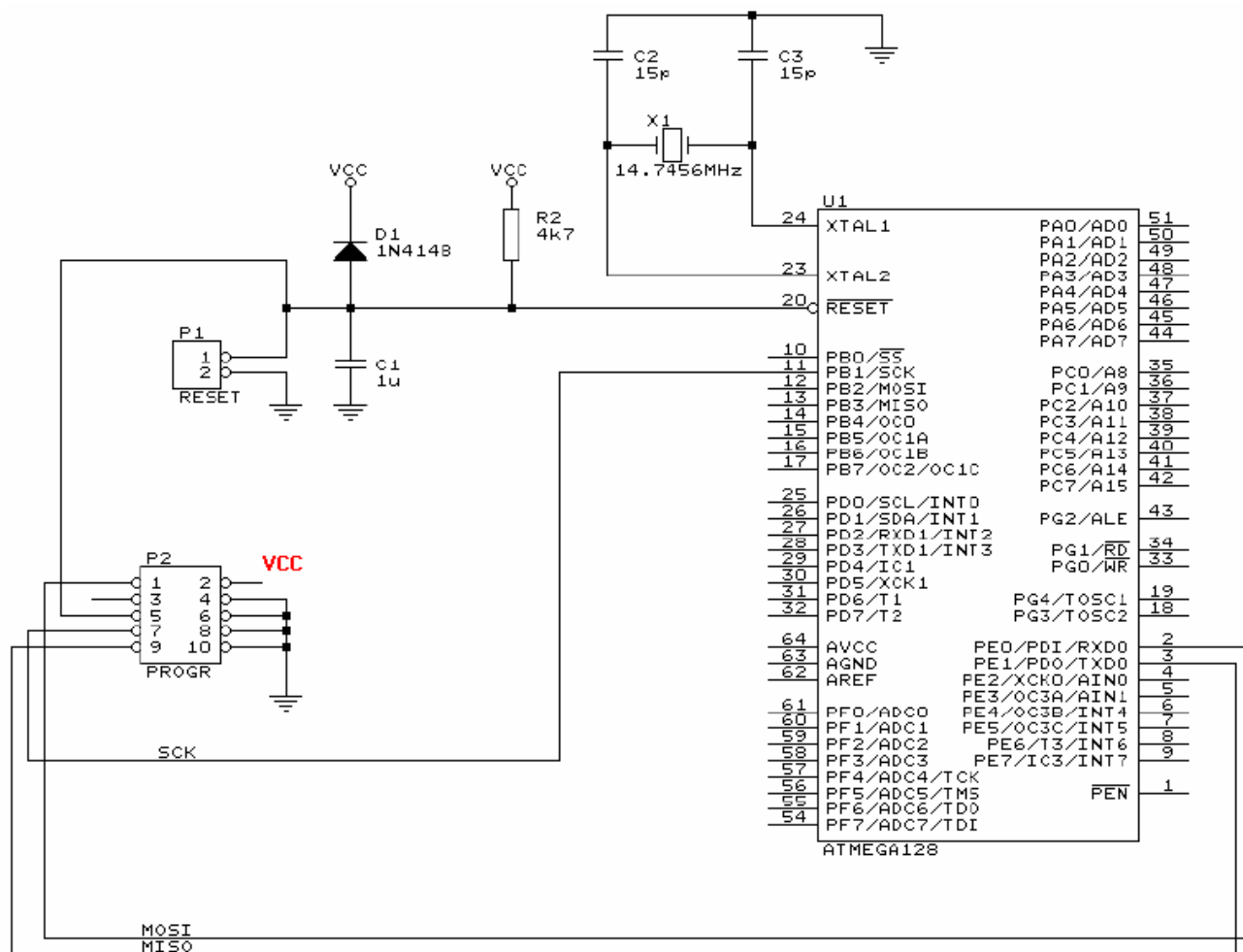


Przykład przyłącza do programatora dla mikrokontrolera Atmega161.



Przykład przyłącza do programatora dla mikrokontrolera Atmega163.

W większości przypadków podłączenie programatora do mikrokontrolera nie odbiega od podstawowej koncepcji, lecz istnieją od tej reguły **WYJĄTKI** ! Przykładem może być mikrokontroler Atmega128, co widać na poniższym rysunku.



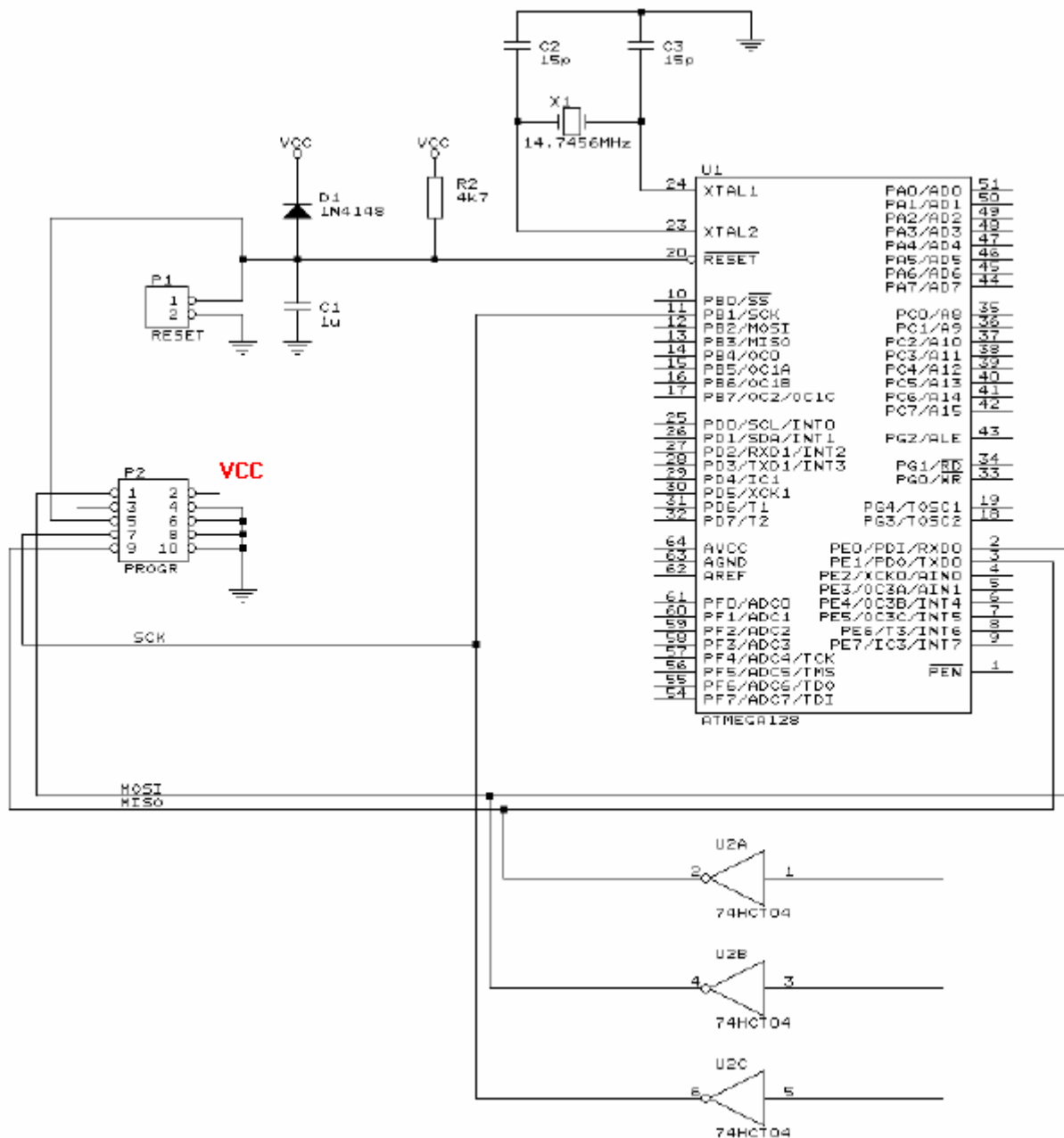
Przykład przyłącza do programatora dla mikrokontrolera Atmega128.

Ze względu na występowanie różnic w podłączeniu niektórych mikrokontrolerów do programatora, należy zawsze sprawdzić w oryginalnej dokumentacji, które wyprowadzenie służy do programowania szeregowego.

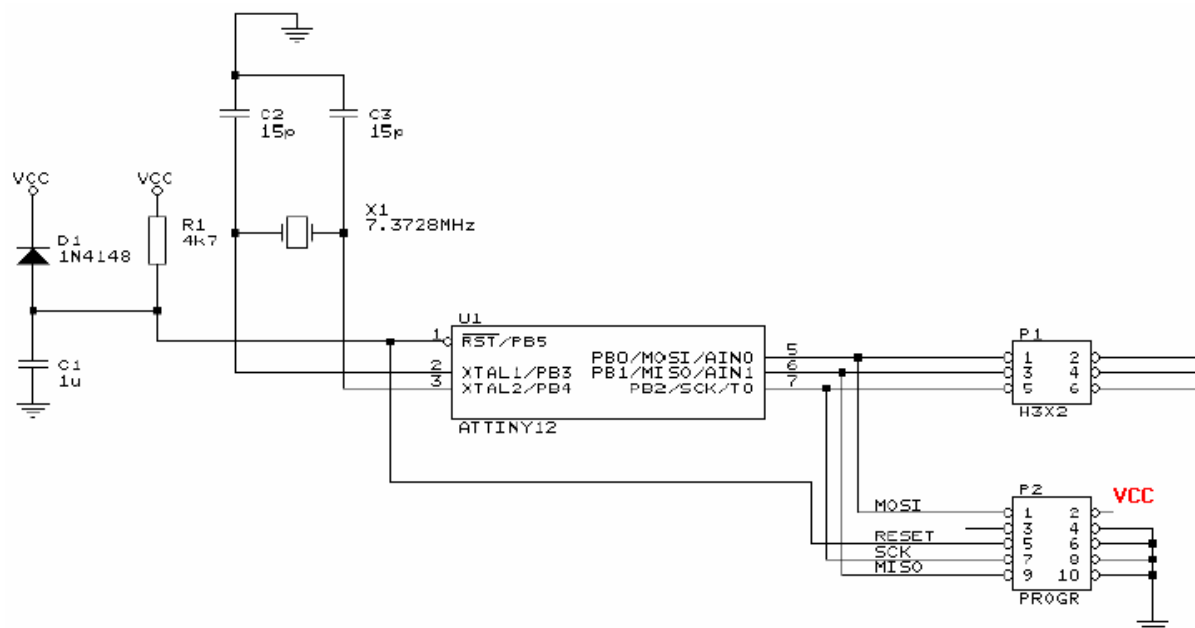
Co zrobić jeśli te same wyprowadzenia mikrokontrolera, które służą do programowania mają także sterować jakimś blokiem cyfrowym w układzie właściwym?

Użycie do programowania sygnałów MOSI, MISO i SCK nie oznacza, że nie mogą one służyć do normalnej pracy mikrokontrolera w układzie docelowym. Jeśli w normalnej pracy mikrokontrolera sygnały te służą zawsze jako wyjście, czyli AVR steruje jakimś blokiem cyfrowym, to nie ma problemu, gdyż podczas programowania wszystkie pozostałe układy podłączone do tych linii są wejściami i nie zakłócają programowania.

Jeśli jednak któryś z pinów służących do programowania, jest użyty w normalnej pracy jako wejście mikrokontrolera, to nie można bezpośrednio podłączyć programatora do tego pinu, gdyż dany układ cyfrowy może wymusić na tej linii własny stan logiczny i programowanie nie będzie możliwe. Sytuacja taka przedstawiona jest na poniższym rysunku.

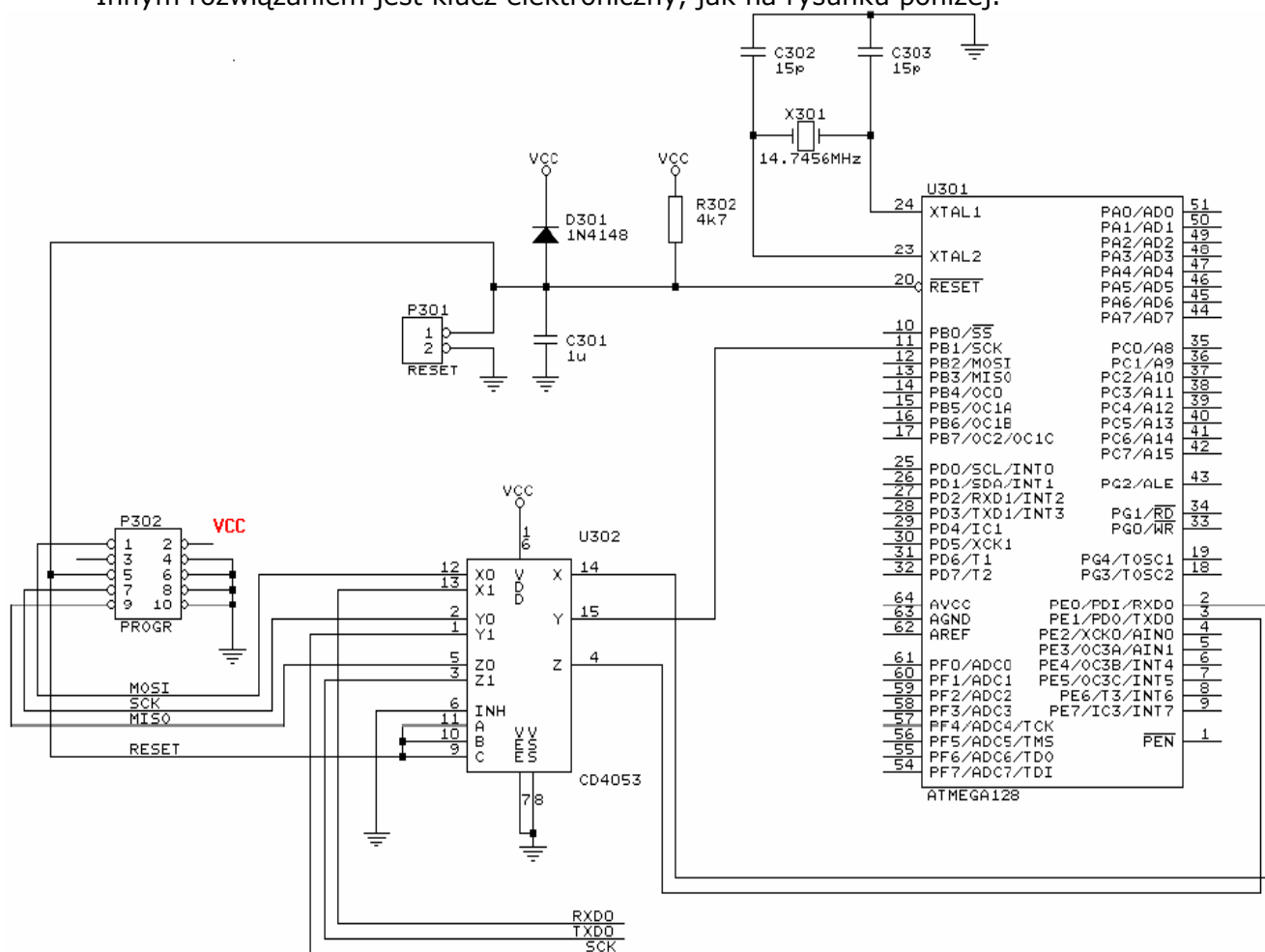


W powyższym układzie powstaje konflikt logiczny między sygnałami programatora i sygnałami wyjściowymi z bramek 74HC04. Rozwiązanie tego typu konfliktów realizuje się na dwa różne sposoby. Pierwszy to fizyczne oddzielenie wybranych wyprowadzeń mikrokontrolera od reszty układu na pomocą zworek, jak pokazuje to rysunek poniżej.



W powyższym układzie należy odłączyć zworki (P1) na czas programowania.

Innym rozwiązaniem jest klucz elektroniczny, jak na rysunku poniżej.



Przykład elektronicznego odłączania konfliktowych sygnałów na czas programowania.

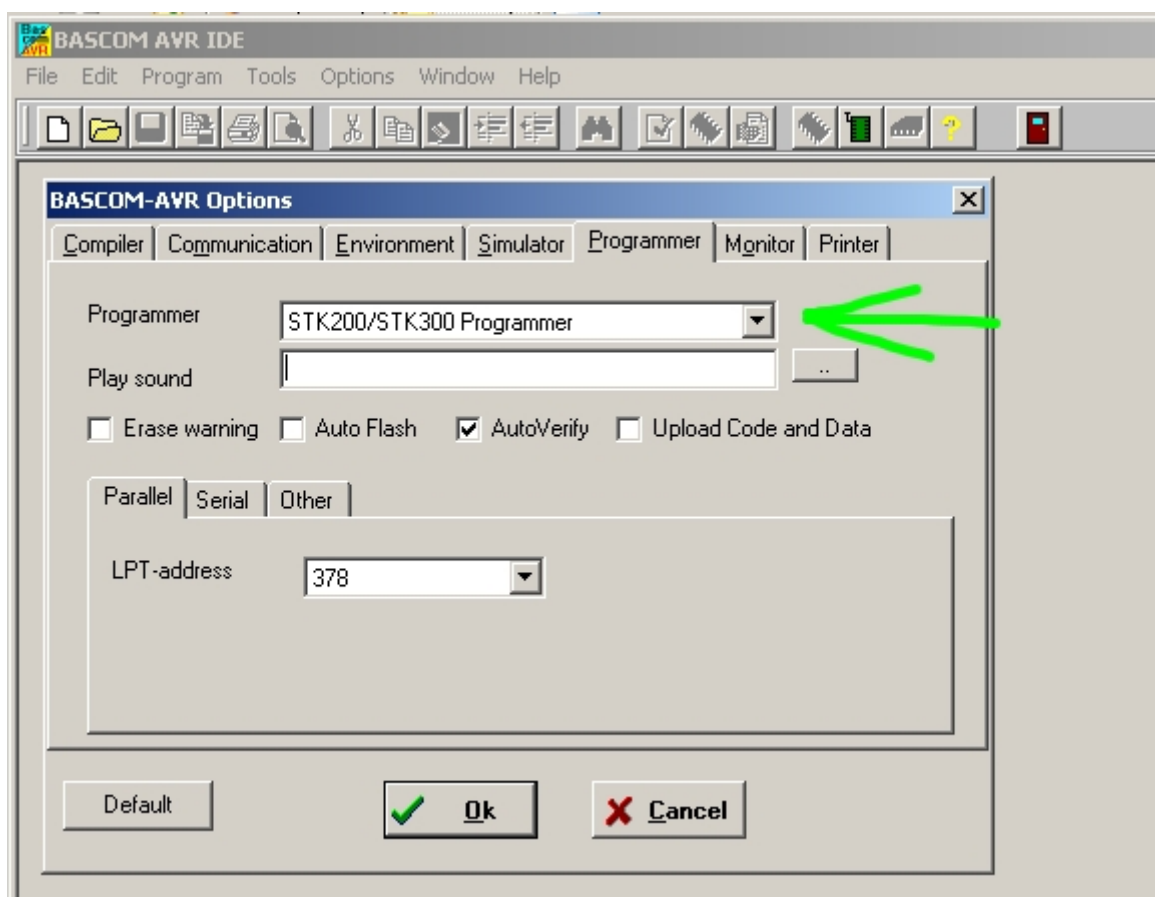
Układ CD4053 przełącza w odpowiedni sposób sygnały używane do programowania szeregowego. Przełączanie sygnałów w czasie programowania odbywa się pod wpływem sygnału RESET, który generowany jest przez programator w trakcie programowania. W

trakcie normalnej pracy układu multiplexer CD4053 łączy mikrokontroler z blokiem cyfrowym układu.

Wszystkie powyższe schematy połączeń programatora do mikrokontrolera mogą być uzupełnione o diodę świecącą LED włączoną między VCC i pin LED (styk nr 3 złącza 10-pinowego), co umożliwi obserwowanie pracy programatora w czasie programowania.

Konfiguracja wybranych programów do pracy z programatorem.

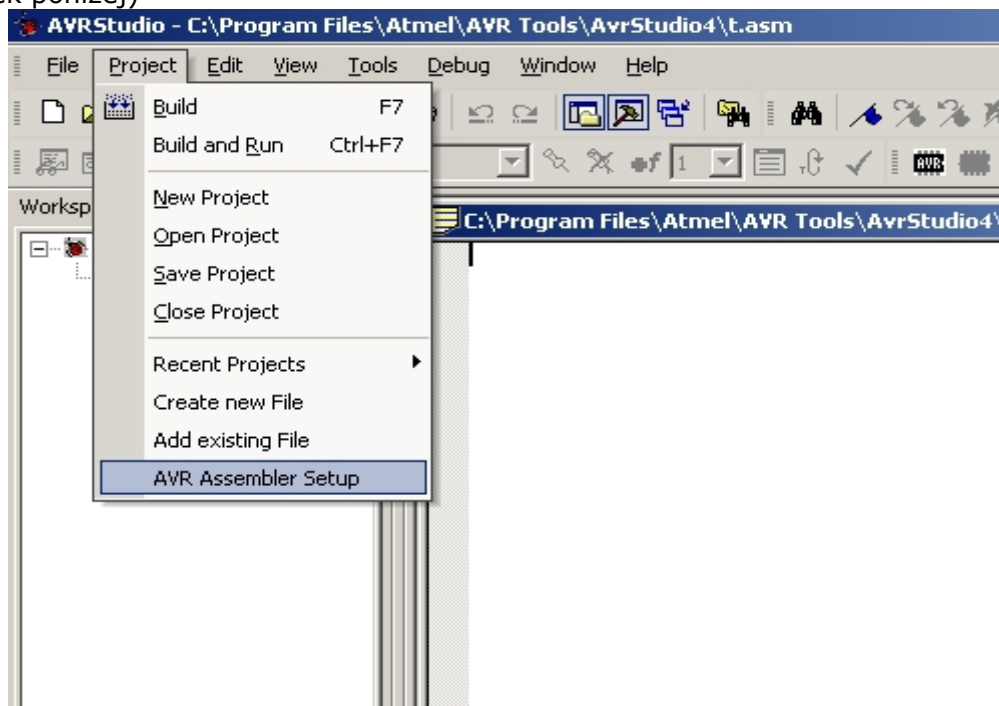
Jeśli używasz programu BASCOM AVR, to należy w opcjach programator ustawić **STK200/STK300 Programmer** jak na rysunku poniżej.



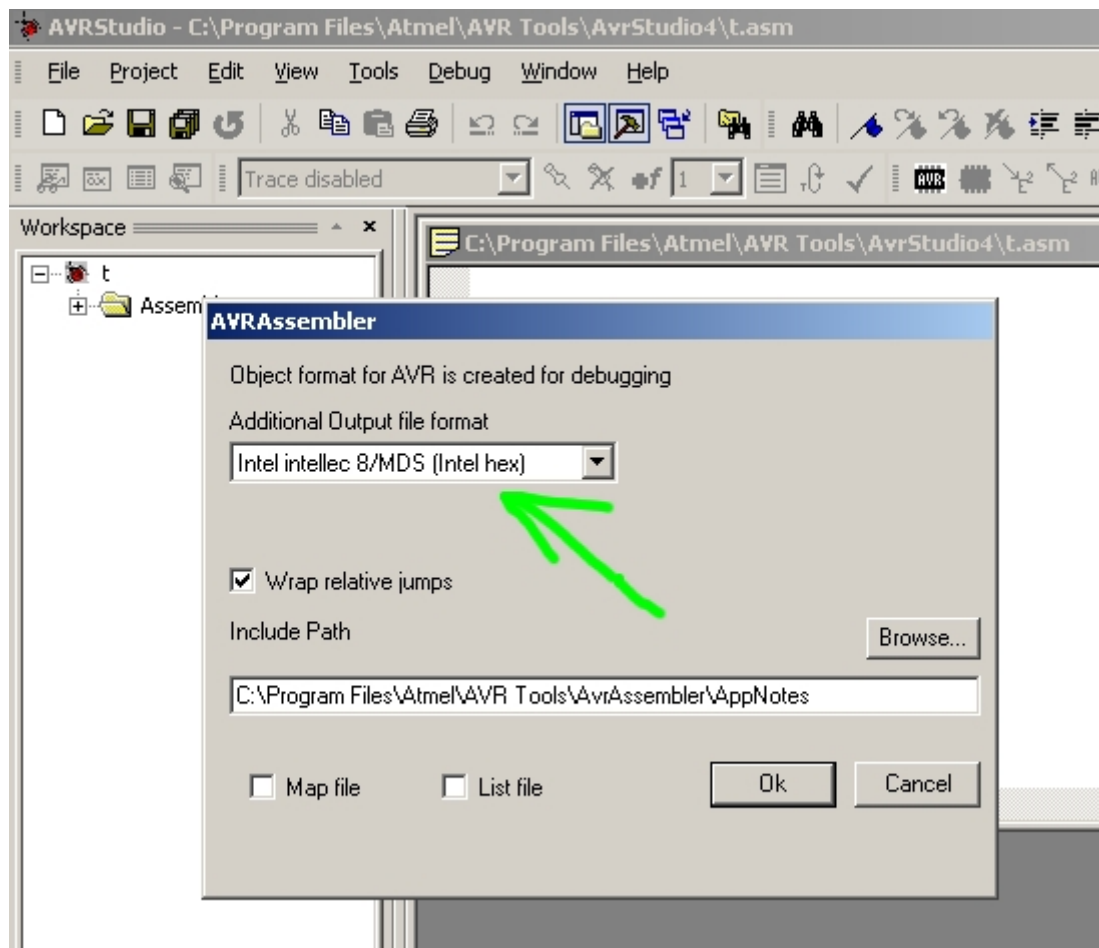
Konfiguracja BASCOM AVR

Programowanie wybranego mikrokontrolera programem ChipBlaster z pliku HEX wygenerowanego przez AVRStudio.

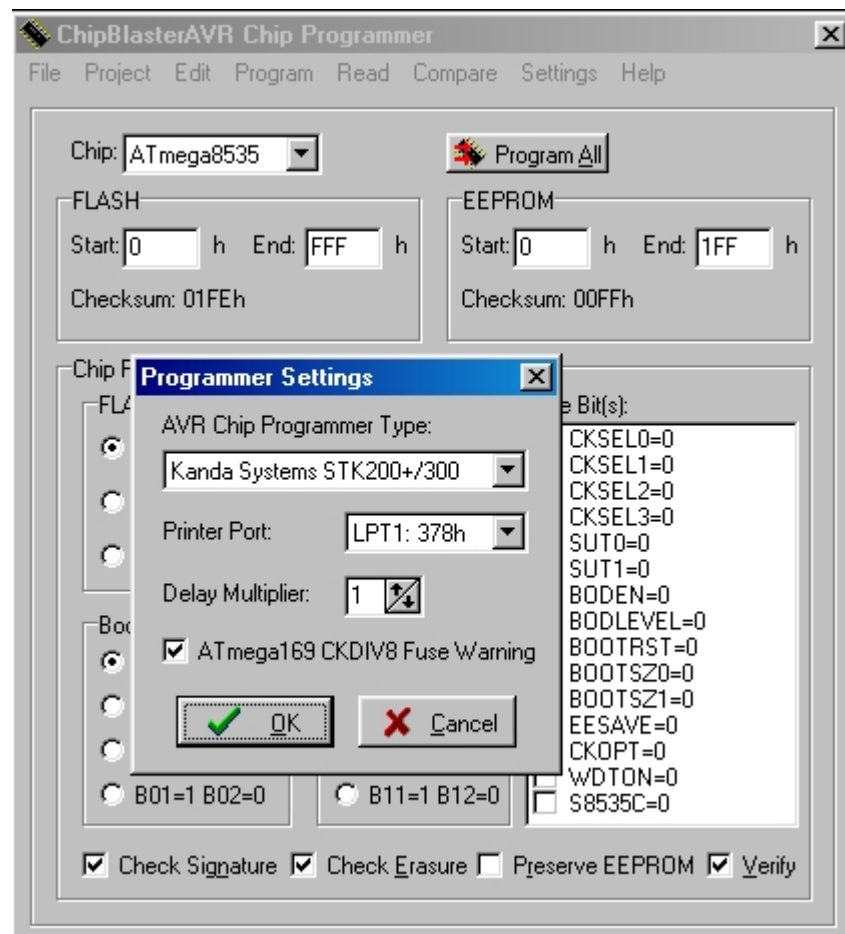
Najpierw przy otwartym projekcie wybierz Project->AVR_Assembler_Setup (rysunek poniżej)



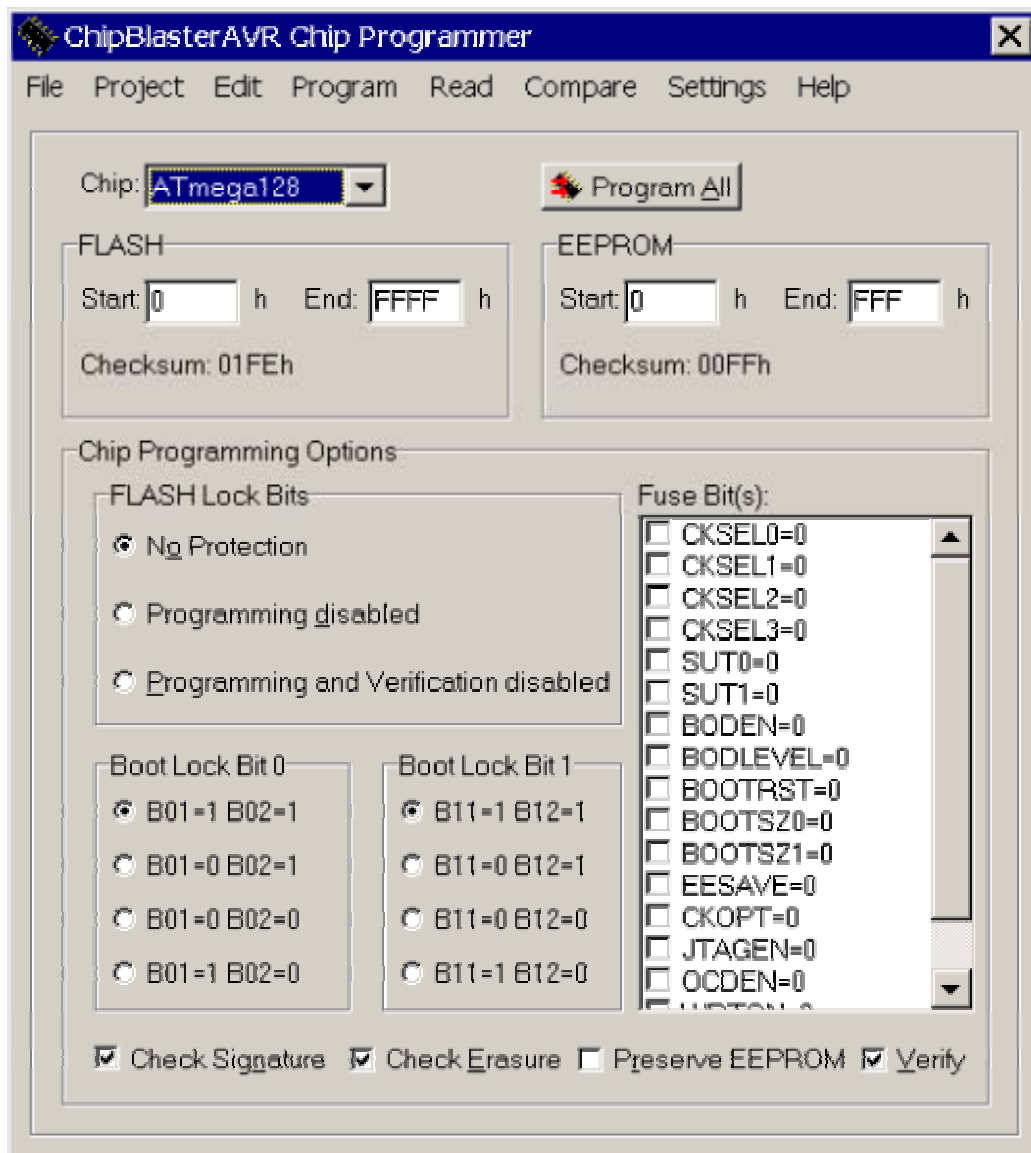
Następnie ustaw opcję „Intel intellec 8/MDS (Intel hex)”, aby plik wyjściowy miał odpowiedni format (rysunek poniżej)



Po skompilowaniu programu w AVRStudio otrzymany plik wyjściowy HEX może być użyty do zaprogramowania wybranego mikrokontrolera AVR. W tym celu otwórz program ChipBlaster i skonfiguruj go jak poniżej.



Następnie podłącz programator do portu równoległego LPT komputera PC i do układu oraz włącz zasilanie. Otwórz programem ChipBlaster wybrany plik HEX i wybierz odpowiedni układ AVR z listy. Od tego momentu można już czytać, zapisywać i weryfikować oraz ustawiać zabezpieczenia w wybranym układzie AVR.



Wersja testowa programu ChipBlaster ogranicza długość kodu. Aby programować AVRy bez ograniczeń można użyć programu PonyProg 2000. Odpowiednia konfiguracja podana jest na rysunku poniżej.

