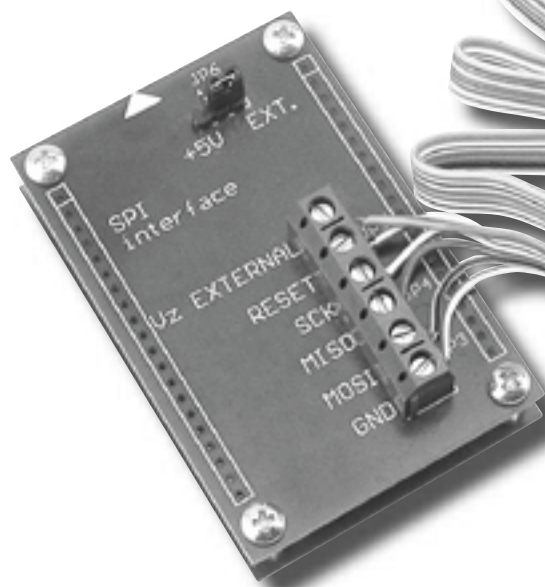


Interfejs SPI do programatora PUNCH

AVT-595

Prezentowana przystawka SPI dla programatora PUNCH, którego projekt przedstawił w EP12/2002 (AVT-5092), służy poszerzeniu możliwości tego narzędzia. Dzięki niej PUNCH jest w stanie programować zamontowane w układzie docelowym różnego typu mikrokontrolery wyposażone w pamięć typu Flash.

Rekomendacje: polecamy obecnym i przyszłym użytkownikom programatora PUNCH, którego funkcjonalność zdecydowanie wzrośnie.



Do tej pory mikrokontrolery z pamięcią Flash mogły być zapisywane i odczytywane wyłącznie w trybie równoległym. Przy takim sposobie programowania zarówno bajt danej, jak i adres określający, do której komórki pamięci Flash dana ma być zapisana, są podawane jednocześnie na odpowiednie porty mikrokontrolera. Wymaga to podłączenia co najmniej kilkunastu sygnałów do wyprowadzeń układu. Wiąże się to z koniecznością umieszczenia mikrokontrolera na czas programowania w odpowiedniej podstawie typu ZIF, a potem przeniesienia do podstawki w urządzeniu docelowym, w którym będzie pracował. Na etapie pisania i poprawiania oprogramowania może to oznaczać nawet kilkaset powtórzeń tych operacji! W dodatku, coraz częściej nowe układy pojawiają się w obudowach przystosowanych do montażu powierzchniowego, co powoduje dodatkowe kłopoty. Po pierwsze, podstawki do programowania elementów w obudowach typu TQFP są drogie. Po drugie, po wlutowaniu zaprogramowanego układu do płytki nie istnieje już praktycznie możliwość jego ponownego przeprogramowania w taki sposób. Z tego powodu producenci mikrokontrolerów wy-

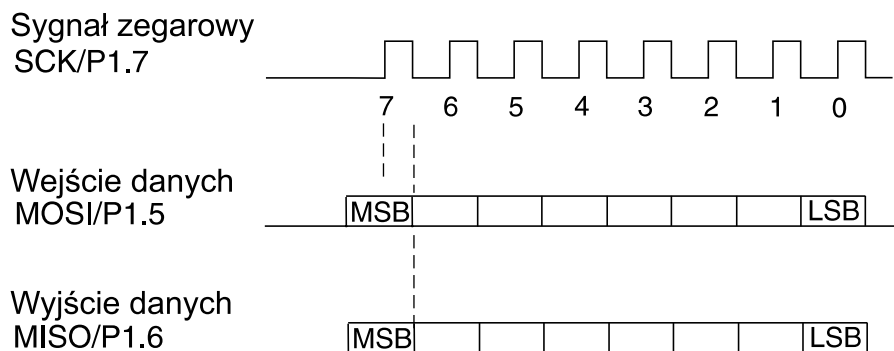
posażają je w funkcję programowania bezpośredniego w układzie, w którym pracują. Programowanie odbywa się za pośrednictwem kilku wyodrębnionych wyprowadzeń mikrokontrolera, które służą do szeregowego przesyłania danych pomiędzy kostką a programatorem. W czasie programowania mikrokontroler pobiera zasilanie z układu, w którym jest zamontowany.

Prezentowana przystawka pozwala programować wybrane mikrokontrolery w tym właśnie trybie. Na początek opracowano i przetestowano metody programowania kilku rodzajów mikrokontrolerów firmy Atmel, m.in.: AT90S2313, AT90S515, ATMEGA-515, AT89S52, AT89S8252, chociaż nic nie stoi na przeszkodzie, by przystawka współpracowała także z innymi układami.

Interfejs SPI Atmela

Wybór mikrokontrolerów Atmela jako pierwszych układów, na których przystawka była testowana, wynika z dużego ich rozpowszechnienia na polskim rynku. Są one łatwo dostępne i stosunkowo tanie.

Zanim przejdziemy do omawiania budowy samej przystawki, warto poznać chociaż ogólne zasady działania interfejsu SPI. Na



Rys. 1. Funkcje wyprowadzeń i sposób transmisji danych przez interfejs SPI

szczęście nie są one zbyt skomplikowane.

Interfejs SPI, służący do programowania w układzie mikrokontrolerów Atmela, składa się z 4 linii sygnałowych. Do tego należy dodać wyprowadzenia masy i napięcia zasilania mikrokontrolera – będzie ono potrzebne w niektórych trybach pracy przystawki. Nazwy i opis funkcji poszczególnych wyprowadzeń przedstawiono poniżej (**rys. 1**):

MOSI – wejście szeregowe danych, tą linią dane są zapisywane do programowanego układu,

MISO – wyjście szeregowe danych, tą linią dane są odczytywane z programowanego układu,

SCK – wejście sygnałów zegarowych, tą linią podawane są do programowanego układu sygnały zegarowe synchronizujące transmisję danych wejściowych i wyjściowych,

RESET – wejście sygnału zerowania mikrokontrolera, podczas programowania układu sygnał ten jest cały czas aktywny.

Jak to wcześniej napisano, podczas programowania mikrokontroler powinien być zasilany, powinien także pracować jego oscylator, co najczęściej oznacza dołączenie do wyprowadzeń XTAL zewnętrznego kwarcu. W czasie programowania układu za pomocą interfejsu SPI podstawowa konfiguracja mikrokontrolera zamontowanego w docelowym układzie powinna wyglądać tak jak na **rys. 2**. Jako przykład pokazano mikrokontroler AT89S52 z sygnałem zerowania aktywnym przy poziomie wysokim. Mikrokontrolery z grupy AVR są zerowane poziomem niskim. Przyporządkowanie linii interfejsu SPI do konkretnych wyprowadzeń poszcze-

gólnych układów można znaleźć w ich dokumentacji technicznej.

Na rys. 1 pokazano przebiegi czasowe sygnałów interfejsu. Dane są przesyłane po liniach MOSI i MISO bajt za bajtem z najstarszym bitem jako pierwszym. Z rysunku wynika, że kolejne bity danych wejściowych na linii MOSI są synchronizowane zboczem narastającym zegara SCK i gdy ma on poziom wysoki, nie powinny się zmieniać. Z kolei dane wyjściowe z mikrokontrolera można odczytywać tuż przed zboczem opadającym zegara SCK. Informacje na liniach MOSI i MISO mogą ulegać zmianie, gdy impuls zegara SCK ma poziom niski.

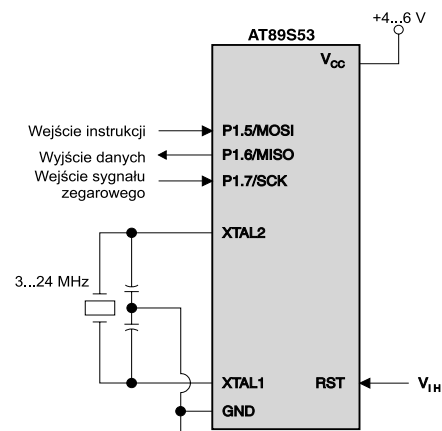
W czasie programowania napięcie zasilania Vcc powinno być stabilne, a jego wartość jest określona w dokumentacji technicznej dla każdego z typów mikrokontrolera. W przypadku niektórych z grupy AVR napięcie zasilania może się mieścić w przedziale 3...5,5 V, dla pozostałych wynosi zazwyczaj 4...5,5 V.

Budowa przystawki

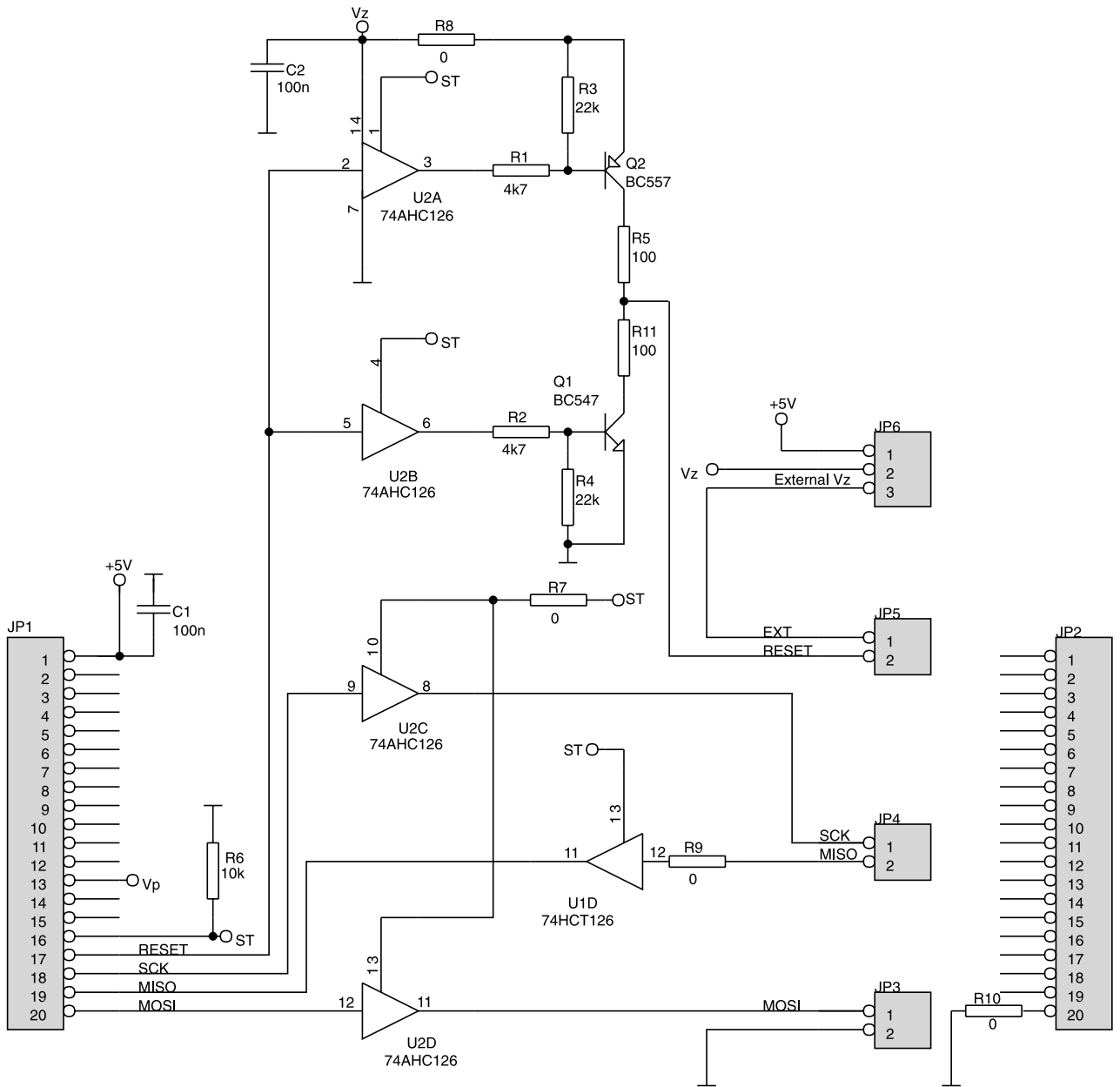
Schemat elektryczny przystawki pokazano na **rys. 3**. Składa się ona z bramek z wyjściem trójstanowym i z klucza tranzystorowego. Zadaniem bramek jest dopasowanie poziomów sygnałów programatora do poziomów akceptowanych przez programowany mikrokontroler oraz odcięcie wszelkich napięć, gdy programowanie dobiegnie końca. Dopasowanie poziomów sygnałów jest istotne w sytuacji, gdy programowany mikrokontroler pracuje w układzie o zasilaniu mniejszym niż +5 V. Jeżeli do budowy przystawki zastosujemy bramki z serii AHC lub LVHC, to uzyskamy niezbędną konwersję poziomów. Bramki te mają bowiem miłą ce-

chę – przy zasilaniu napięciem o wartości od +3 do +5 V tolerują sygnały wejściowe o poziomie przewyższającym poziom napięcia zasilania. Jeżeli dokładnie obejrzymy schemat, to okaże się, że bramki U2A-D zasilane są napięciem Vz pobieranym ze styku 2 złącza JP6. Zależnie od potrzeb napięcie to może być pobierane z programatora i mieć wartość +5 V lub z układu, w którym pracuje programowany mikrokontroler, czyli ze styku JP6-3 oznaczonego jako *External Vz*. O źródle zasilania decyduje zwarcie odpowiednich styków JP6. Jeżeli układ, w którym pracuje mikrokontroler, jest zasilany napięciem +3,3 V, to taką samą wartość będzie miało napięcie *External Vz* zasilające bramki U2A-D. Pomimo że sygnały z programatora doprowadzone do wejścia tych bramek mają poziom TTL, na ich wyjściach sygnały będą miały wartości 0/3,3 V, czyli w pełni akceptowaną przez programowany mikrokontroler. Jedynym wyjątkiem jest bramka U1A typu HCT – na stałe zasilana +5 V. Bramka ta pełni rolę konwertera poziomu sygnału odczytywanego z programowanego mikrokontrolera. Ponieważ bramki z rodziny HCT prawidłowo współpracują z sygnałami z przedziału 2...5 V, sprawdzą się także w roli pośrednika pomiędzy programatorem a mikrokontrolerem zasilanym w przykładowym układzie napięciem +3,3 V.

Bramki sterowane są sygnałem ST, mającym podczas programowania poziom wysoki, który je otwiera. Po zakończeniu cyklu programowania sygnał ST będzie miał poziom niski, co na wyjściach



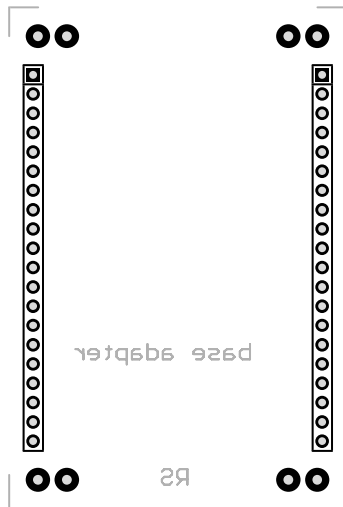
Rys. 2. Konfiguracja mikrokontrolera AT89Sxx podczas programowania ISP



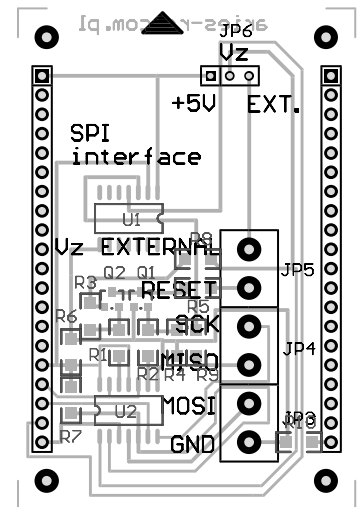
Rys. 3. Schemat elektryczny przystawki

bramek wymusi stan wysokiej impedancji. Dzięki temu wyeliminowane zostają stany nieustalone w czasie dołączania i odłączania programatora mogące grozić uszkodzeniem programowanego elementu.

Klucz tranzystorowy Q1 i Q2 służy do polepszenia parametrów sygnału zerowania podawanego na wejście RESET mikrokontrolera. Zależnie od typu programowanego elementu tranzystory mogą wymuszać na wejściu RESET poziom wysoki lub niski. Po zakończeniu programowania, gdy bramki U2A i U2B są zamykane, oporniki tak polaryzują bazy tranzystorów, aby i one były odcięte i nie obciążały wprowadzenia RESET.



Rys. 4. Schemat montażowy płytki dolnej



Rys. 5. Schemat montażowy płytki górnej

Montaż

Przystawka składa się z dwóch jednostronnych płytek drukowanych (schematy montażowe pokazano na **rys. 4** i **5**) skręconych ze sobą w postać „kanapki” za pomocą kołków dystansowych. Do dolnej płytki są wlotowane dwa jednorzędowe grzebienie po 20 styków każdy, przeznaczone do mocowania przystawki w gnieździe programatora PUNCH. Na górnej płytce znajduje się właściwy układ przystawki. Ponieważ do budowy zostały użyte elementy SMD, należy je przylutować do płytki od strony druku. Jeżeli przystawka będzie wyłącznie służyć do programowania mikrokontrolerów zasilanych napięciem +5 V, można w miejsce układu U2 wlotować jego łatwiej dostępną wersję 74HCT126. Zdziwionych obecnością na schemacie i płytce elementów oznaczonych jako oporniki o wartości 0Ω chciałbym uspokoić – to nie pomyłka. Ponieważ ze względów ekonomicznych została użyta płytka jednostronna, elementy te pełnią po prostu rolę zworek umożliwiających właściwe poprowadzenie wszystkich ścieżek.

Na górnej części tej płytki znajdują się gniazda JP3, 4 i 5 dla przewodów łączących programator z programowanym mikrokontrolerem oraz styki JP6 pola wyboru napięcia zasilania układu. Dolną i górną płytkę przystawki należy połączyć przewodami w ten sposób, aby styki złącza JP1 oznaczone na schemacie symbolami MISO, MOSI, SCK, RESET i ST łączyły się z sygnałami w gnieździe programatora oznaczonymi jako F1...F5. Rozkład wyprowadzeń F1...F5 w gnieździe programatora można znaleźć, uruchamiając program sterujący PUNCH-a i przechodząc do pulpitu serwisowego. Oprócz tego trzeba pamiętać o połączeniu ze sobą styków JP1-1 i JP2-20 dolnej i górnej płytki. Jako połączenie pomiędzy programatorem a programowanym mikrokontrolerem może służyć 6-przewodowa taśma o długości do 40 cm. Przewody z jednej strony trzeba zamocować w gniazdach JP3-5 np. zaciskowych typu ARK, z drugiej taśma powinna być zakończona wtykiem dla złącza w układzie, w którym znajduje się programowany mikrokontroler.

Projektowanie urządzeń przystosowanych do programowania w systemie

Aby mikrokontroler mógł być zaprogramowany w układzie, urządzenie, w którym pracuje, musi być odpowiednio zaprojektowane. Najwygodniej jest, gdy wyprowadzenia MISO, MOSI i SCK programowanego mikrokontrolera nie są połączone z innymi elementami układu lub gdy takie połączenie na czas programowania można przerwać. Jeżeli nie jest to możliwe, należy przynajmniej zadbać, aby wyprowadzenia nie były specjalnie obciążane dołączonymi elementami. Najlepiej jeśli wyprowadzenia pełnią funkcję wyjść łączonych z wejściami np. innych bramek. Jeśli w czasie programowania na wyprowadzeniach portu SPI zaczną pojawiać się sygnały z układu, w którym pracuje mikrokontroler, to programowanie zostanie zakłócone i ulegnie przerwaniu. Należy o tym pamiętać, gdyż mikrokontroler programowany jest we włączonym urządzeniu, z którego przecież pobiera energię.

Kolejnym problemem jest wejście RESET, a właściwie dołączone do tego wejścia układy zerujące mikrokontroler. Na czas programowania powinny zostać odcięte. Wyjątkiem są niewielkie kondensatory, które często bywają wykorzystywane do wytwarzania impulsu zerującego po włączeniu zasilania. Mają one na tyle małą pojemność, że układ klucza przystawki powinien sobie z nimi poradzić.

Bardzo wygodne jest umieszczenie na płytce drukowanej urządzenia, w którym pracuje programowany mikrokontroler, grzebienia z wyprowadzonymi liniami portu SPI. Ułatwi to dołączanie programatora i jest lepszym rozwiązaniem niż każdorazowe lutowanie przewodów do odpowiednich wyprowadzeń mikrokontrolera.

Korzystanie z przystawki

Po zmontowaniu przystawka od razu nadaje się do pracy, należy włożyć ją do gniazda programatora i połączyć jej wyjścia z odpowiednimi wyprowadzeniami mikrokontrolera. Przystawka jest przystosowana do pracy z mikrokontrolerami pobierającymi zasilanie z układu, w którym pracują.

Przed rozpoczęciem programowania należy doprowadzić napięcie zasilania procesora do styku złącza JP5 oznaczonego jako Vz *EXTERNAL*. Następnie należy zewrzeć ze sobą wyprowadzenia 2 i 3 złącza JP6 opisane jako Vz i EXT. Po włączeniu zasilania w urządzeniu, w którym znajduje się programowany mikrokontroler i wybraniu odpowiedniego skryptu, można przystąpić do programowania.

Jeżeli układ, w którym pracuje mikrokontroler, potrzebuje do zasilania niewielkiego prądu rzędu 80...100 mA, to napięcie do zasilania procesora o wartości +5 V może być doprowadzone bezpośrednio z programatora. W takim przypadku w czasie programowania urządzenie, w którym znajduje się mikrokontroler, nie musi być włączone. Tak jak poprzednio należy połączyć wyprowadzenie końcówki Vcc programowanego elementu ze złączem JP5 oznaczonym jako Vz *EXTERNAL*. Następnie należy połączyć zwrora wszystkie trzy wyprowadzenia złącza JP6, co po rozpoczęciu programowania umożliwi podanie na mikrokontroler napięcia +5 V. Skrypty napisano w taki sposób, aby po podaniu przez programator zasilania wystąpiła pauza ok. 1 s, co zazwyczaj wystarcza do przeładowania pojemności filtrujących w urządzeniu z mikrokontrolerem i do ustabilizowania się zasilania. *Nigdy nie należy jednocześnie doprowadzać zasilania z programatora i włączać własnego zasilacza urządzenia!*

Skrypty

Użytkownicy programatora PUNCH doskonale wiedzą, że do zaprogramowania elementu potrzebny jest skrypt będący rodzajem instrukcji dla programatora. Do projektu przystawki dołączono kilka skryptów pozwalających programować różne wymienione wcześniej mikrokontrolery. Skrypt jest plikiem tekstowym i każdy zainteresowany może podejrzeć jego strukturę, korzystając ze zwykłego edytora tekstu. Pisanie skryptu dla nowego elementu bywa zadaniem uciążliwym, ale można sobie ułatwić życie, korzystając z już napisanych skryptów. Zainteresowanych zapraszam na swoją stronę www.aries-rs.com.pl/femto, gdzie można znaleźć bezpłatne skrypty do pro-

WYKAZ ELEMENTÓW**Rezystory i kondensatory
w obudowach typu 1206****Rezystory**

R1 R2: 4,7kΩ

R3, R4: 22kΩ

R5, R11: 100Ω

R6: 10kΩ

R7, R8, R9, R10: zworki

U2: 74AHC126

U1: 74HCT126

Kondensatory

C1, C2: 100nF

Półprzewodniki

Q1: BC547 lub dowolny tranzystor typu NPN w obudowie do montażu powierzchniowego

Q2: BC557 lub dowolny tranzystor typu PNP w obudowie do montażu powierzchniowego

gramowania różnych elementów, opis języka skryptów FEMTO oraz narzędzia do pisania skryptów, czyli edytor *Skryptorium* (wszystkie te materiały publikujemy także na CD-EP10/2004B).

W niektórych przypadkach potrzeba bardzo niewiele czasu, aby

przystosować już istniejący skrypt do nowego elementu. Jako pierwszy „z brzegu” przykład niech posłużą mikrokontrolery AT90S2313 i AT90S8515. Wystarczy ze strony producenta firmy ATMEL ściągnąć dokumentację techniczną obydwu mikrokontrolerów i porównać te fragmenty, które dotyczą sposobu programowania – są identyczne. Jedyna różnica polega na rozmiarze dostępnej pamięci programu i pamięci EEPROM. Po małej modyfikacji ten sam skrypt służy do programowania obydwu elementów. Co bardziej dociekliwi Czytelnicy łatwo mogą go odszukać na stronie, a potem przeanalizować strukturę. W podobny sposób można rozbudować o nowe elementy skrypty napisane dla elementów programowanych za pomocą przystawki SPI.

Niezbędna nowa wersja oprogramowania PUNCH-a

Na koniec istotna uwaga: do prawidłowego działania przystawki, a zwłaszcza skryptów sterujących programowaniem, niezbędne jest,

aby program procesora PUNCH miał numer 2.1 lub 3.1. Numer wersji oprogramowania wyświetla się w okienku oznaczonym jako „wersja czytnika” znajdującym się na pulpicie serwisowym. W trybie serwisowym wystarczy jedna transmisja, żeby w okienkach informacyjnych pojawił się numer programatora i wersja jego oprogramowania. Pojawienie się numeru niższego niż 2.1 oznacza konieczność unowocześnienia oprogramowania (nową wersję programu sterującego publikujemy na CD-EP10/2004B). Na wszelki wypadek proponuję zachować mikrokontroler ze starą wersją programu, a nową wersję oprogramowania wgrać do pamięci innego mikrokontrolera AT89C52 lub AT89S52. W razie problemów proszę o kontakt na adres poczty elektronicznej.

Ryszard Szymaniak, EP
ryszard.szymaniak@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: pcb.ep.com.pl oraz na płycie CD-EP10/2004B w katalogu PCB.