

Myszka komputerowa dla osób niepełnosprawnych, część 2

AVT-862



W drugiej części artykułu omawiamy oprogramowanie mikrokontrolera „myszki” oraz jej montaż i uruchomienie.

Za miesiąc przedstawimy w EP program „wirtualnej klawiatury”, którego zastosowanie ułatwi korzystanie z komputera osobom niepełnosprawnym.

Oprogramowanie

Program dla mikroprocesora napisałem w języku C. Do kompilacji używałem produktu firmy ImageCraft (www.imagecraft.com/software), noszącego nazwę IC-CAVR. Z czystym sumieniem mogę polecić ten kompilator amatorom procesorów AVR. Prosta obsługa pozwala rozpocząć prace natychmiast po zainstalowaniu programu. Doskonale została rozwiązana diagnostyka błędów rozpoznawanych na etapie kompilacji i linkowania. Oprócz plików do programowania pamięci programu typu *.hex*, kompilator może generować pliki *.cof*, które są akceptowane przez symulator AVR-Studio. Dzięki temu wstępne uruchamianie można prowadzić na poziomie kodu źródłowego, co znacznie przyspiesza pracę i poprawia jej komfort. Należy do tego dodać dobrą jakość generowanego kodu i dostęp do postaci źródłowej bibliotek standardowych.

Program sterujący składa się z siedmiu części. Na początku procesor inicjalizuje stos i zmienne. Zaraz potem następuje odesła-

nie do komputera identyfikatora myszy. Pięć kolejnych bloków stanowi główną pętlę programu. Dwa pierwsze z nich są niemal identyczne i realizują odczyt wyjść akcelerometrów. W kolejnym, program interpretuje trójfazowy przebieg na wyjściu przetwornika UTI. Procesor musi też sprawdzić, czy nie zostały zwarte opcjonalne przełączniki, odpowiadające lewemu i prawemu klawiszowi myszki. Na końcu to wszystko jest upychane w jednym, trzypobytowym komunikacie, wysyłane do komputera i program wraca na początek pętli.

Na pierwszy rzut oka program wydaje się dość prosty. W rzeczywistości napisanie i uruchomienie zajęło mi tyle czasu, że 30-dniowa wersja demonstracyjna programu kompilatora przestała działać i zostałem niejako zmuszony do zakupu jego pełnej wersji.

O właściwą inicjalizację zmiennych i stosu musi zadbać sam kompilator, zatem program rozpocząłem od funkcji realizującej zgłoszenie komputerowi obecności myszki. W standardzie Microsoft Serial Mouse myszka zgłasza się

wysyłając literę „M“ (77 ASCII) po każdym ujemnym impulsie na linii RTS (Z7). Transmisja odbywa się z szybkością 1200 bodów. Nadawane słowo zawiera bit startu, siedem bitów danych i dwa bity stopu. Identyfikator ma zatem postać ciągu bitów: 0_1110111_11. Procesor AT90S2313 jest wyposażony w port szeregowy. Dzięki temu cała procedura identyfikacji sprowadziła się do ustawienia parametrów transmisji, załadowania bufora i czekania na sygnał końca nadawania.

Kolejnym zadaniem programu jest odczyt sygnałów z akcelerometrów. Wyjścia obu akcelerometrów są podłączone do wejść przerwania zewnętrznych INT0 (6, U1) i INT1 (7, U1). Program najpierw ustawia wejście INT0 jako czułe na zbocze opadające i wykonywanie programu zostaje wstrzymane instrukcją SLEEP. Gdy na wejściu pojawi się takie zbocze, procesor budzi się i przechodzi do obsługi przerwania. Ta i wszystkie pozostałe procedury obsługi przerwania są puste. Dzięki temu procesor szybko wraca do programu głównego. Teraz jest uruchamiany wewnętrzny, 16-bitowy Timer1. Wejście INT0 uczulane jest na zbocze narastające i procesor jest ponownie usypiany. Zbocze kończące impuls z akcelerometru budzi procesor, który zatrzymuje Timer1 i zapamiętuje jego zawartość. W ten sposób mierzony jest czas trwania ujemnego impulsu na wyjściu oznaczonym przez producenta jako OUTX (10, U2). W naszym układzie jest to akurat wyjście akcelerometru czułego na pochylenie głowy do przodu i do tyłu, czyli w osi Y.

Fragment programu realizujący ten pomiar można prześledzić

na list. 1. Zainteresowanym Czytelnikom chciałbym zwrócić uwagę na sposób wstawiania instrukcji asemblerowych, bo jest to kolejna zaleta kompilatora IC-CAVR.

Taki sam cykl powtarza się dla akcelerometru OUTY (9, U2), podłączonego do wejścia INT1. Różnica polega tylko na tym, że mierzony jest czas trwania impulsu dodatniego.

Tutaj należy się kilka słów wyjaśnienia. Zasadniczo, aby móc obliczyć wartość przyspieszenia, konieczne jest mierzenie dwóch parametrów impulsów: czasu trwania impulsu i okresu powtarzania całego przebiegu. Dopiero iloraz tych czasów wskazuje dokładny wynik pomiaru. W naszym przypadku taka dokładność nie jest konieczna. Z dobrym przybliżeniem możemy założyć, że okres przebiegu na wyjściu akcelerometrów jest stały. Taki stały czynnik można pominąć, gdyż nie interesuje nas bezwzględna wartość przyspieszenia, a tylko zmiana tej wartości. Zależnie od tego, czy będziemy mierzyć impulsy ujemne, czy dodatnie, wzrost przyspieszenia będzie powodował zwiększenie lub zmniejszanie współrzędnych kursora na ekranie.

Aby zmniejszyć błędy wynikające z bramkowania, zawartość licznika jest dzielona przez cztery. Dla zachowania rozdzielczości konieczne było podwyższenie częstotliwości taktującej. Wybrałem kwarc o „telewizyjnej“ częstotliwości 3,58MHz.

Wcześniej sygnalizowałem problem szumów na wyjściach akcelerometrów. Nie chodzi o szumy w tradycyjnym rozumieniu tego słowa. „Szumienie“ akcelerometrów objawia się losową zmianą długości im-

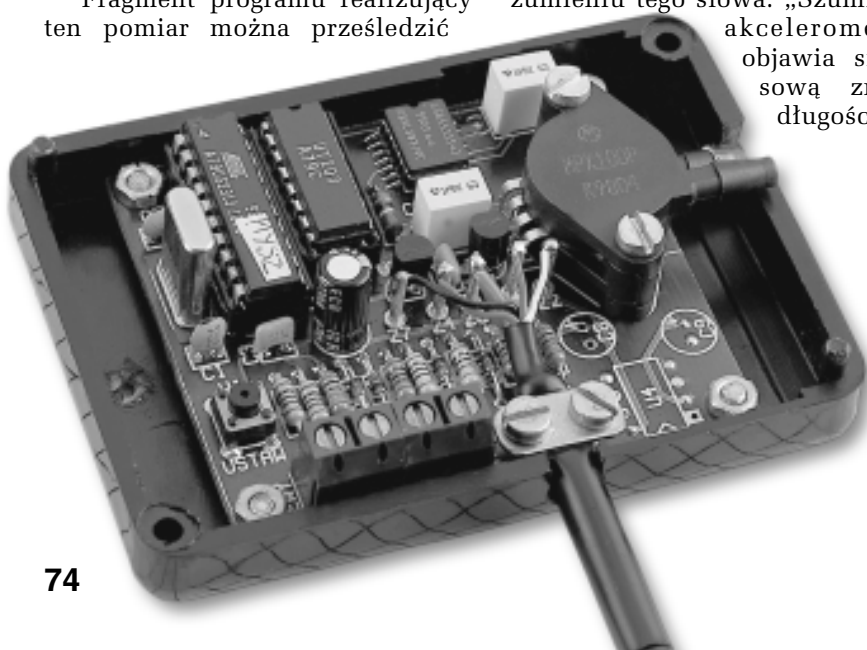
List.1. Podprogram realizujący pomiar przechylenia w osi Y.

```

/** Pomiar przechylenia w osi Y ***/
GIMSK=0x00; /* INT0: wyłącz */
MCUCR=0x22; /* INT0: 1->0 */
GIMSK=0x40; /* INT0: włącz */
asm("SLEEP"); /* czekaj na 1->0 */
asm("NOP");
GIMSK=0x00; /* INT0: wyłącz */
TCCR1B=0x00; /* TIMER1: stop */
TCNT1=0x0000; /* TIMER1: zeruj */
TCCR1B=0x01; /* TIMER1: CK / 1 */
MCUCR=0x23; /* INT0: 0->1 */
GIMSK=0x40; /* INT0: włącz */
asm("SLEEP"); /* czekaj na 0->1 */
asm("NOP");
GIMSK=0x00; /* INT0: wyłącz */
TCCR1B=0x00; /* TIMER1: stop */
probkiY[lprob]=TCNT1; /* nowy czas */
sumaY=0; /* sumuj próbki */
for(i=0;i<LPROB;i++)
  sumaY+=probkiY[i];
deltaY=(sumaY-psumaY)/DZIEL;
if(deltaY>127)
  deltaY=127; /* tyle można wysłać */
if(deltaY<-128)
  deltaY=-128;
psumaY=sumaY;
    
```

pulsów wyjściowych. Program zmniejsza efekty tego zjawiska przez uśrednianie wyników z ośmiu ostatnich pomiarów. Powoduje to wprawdzie dodatkowe opóźnienie reakcji myszki na ruchy głową, ale wyszedłem z założenia, że ważniejsza jest możliwość precyzyjnego ustawienia kursora.

Nieco bardziej rozbudowany jest fragment programu odczytujący przebieg wyjściowy z przetwornika UT1. Przebieg ten składa się z trzech faz, różniących się czasem trwania. Kształt przebiegu ilustruje rys. 4. Pierwsza faza, oznaczona jako *Toff*, wskazuje offset toru pomiarowego. Dla wyróżnienia składa się ona z dwóch identycznych okresów. Druga faza - *Tab* - służy do pomiaru napięcia zasilającego czujnik. Ostatnia, trzecia faza, określa napięcie niezrównoważenia mostka pomiarowego i jest oznaczona jako *Tcd*. Ze względu na ściśle czasowy charakter przebiegu, tutaj również wykorzystałem timer T1 z tym, że licznik zlicza impulsy bez zatrzymywania. Sygnał wyjściowy z przetwornika UT1 jest podawany do procesora na wyprowadzenie o nazwie ICP - Input Capture Pin (11, U1). Przy odpowiednim skonfigurowaniu tego wejścia, każde pojawiające się na nim narastające zbocze powoduje wygenerowanie przerwania i równoczesne przepisanie aktualnej zawartości licznika T1 do specjalnego rejestru ICR. Poprzednia zawartość licznika jest odejmowana od aktualnej i ta różnica jest cyklicznie zapisywana do 4-elementowej tablicy. Po każdym przerwaniu procesor sprawdza, czy pierwsze dwa elementy



spełniają warunki nałożone na fazę *Toff*. Jeśli tak, procesor może wyliczyć wartość ciśnienia ze wzoru:

$$P = 1/32 * (Tcd - Toff) / (Tab - Toff)$$

Podobnie jak w przypadku akcelerometrów, tutaj również nie jest nam potrzebna bezwzględna wartość ciśnienia. Wystarczy, jeśli będziemy sprawdzać, czy naciśnięcie (lub podciśnienie) nie przekracza pewnej wartości. W tym celu procesor musi znać spoczynkową wartość (*Tcd-Toff*). Taki wzorcowy pomiar jest dokonywany jeden raz, a wynik zostaje zapisany w pamięci nieulotnej EEPROM. Sposób przeprowadzenia kalibracji zostanie opisany nieco później. Przy kolejnych pomiarach aktualna wartość (*Tcd-Toff*) jest porównywana z wartością odniesienia. Jeśli obliczone w ten sposób naciśnięcie przekracza wartość minimalną, jest to interpretowane jako naciśnięcie lewego przycisku myszki. Jeśli natomiast w czujniku pojawi się podciśnienie o odpowiednio dużej wartości, program przyjmie, że naciśnięto przycisk prawy.

Niezależnie od pomiaru ciśnienia, procesor sprawdza stan dwóch pinów PB2 (14, U1) i PB4 (16, U1). Wejścia te mają włączone wewnętrzne podciągnięcie do plusa zasilania i w spoczynku występuje na nich poziom wysoki. Jeśli teraz zewrzymy wyprowadzenia złącza Z1, na wejściu PB4 pojawi się poziom niski i procesor odczyta to jako naciśnięcie lewego przycisku. Analogicznie zwarcie złącza Z6 zostanie potraktowane jako naciśnięcie prawego klawisza myszki.

Programowa obsługa przycisku USTAW (W1) jest realizowana w dwóch blokach programu. Pierwszy raz jego stan jest sprawdzany na początku programu, na etapie inicjalizacji. Jeśli procesor stwierdzi, że styki tego przycisku są zwarte, blokuje odczytywanie przetwornika UTI. W takim przypadku jedynym sposobem naciśnięcia klawiszy myszki jest zwiernie złącza Z1 i Z6.

Podczas normalnej pracy stan mikroprzełącznika W1 jest sprawdzany po każdym pomiarze ciśnienia. Zwarcie styków w takim momencie powoduje zatrzymanie programu i zapis aktualnego ci-

Tab.2. Struktura komunikatu w standardzie MS-Mouse.

bajt/bit	7	6	5	4	3	2	1	0
1	1	1	LB	RB	Y7	Y6	X7	X6
2	1	0	X5	X4	X3	X2	X1	X0
3	1	0	Y5	Y4	Y3	Y2	Y1	Y0

nienia do pamięci nieulotnej. Tak zapamiętana wartość jest wartością odniesienia przy interpretowaniu następnym pomiarów. Po zwolnieniu przycisku program kontynuuje działanie z nową zawartością pamięci EEPROM.

W ostatnim kroku program musi złożyć rezultaty pomiaru przechyliń w obu osiach z wynikami pomiaru ciśnienia. Potem trzeba dodać do tego stan na złączach przycisków dodatkowych i skompletowany w ten sposób komunikat można wysłać do komputera. W standardzie MS-Mouse myszka wysłała do komputera trzy bajty. Pierwszy bajt zawiera informacje o stanie klawiszy i dwa najstarsze bity przesunięcia w poziomie i pionie. W drugim bajcie znajduje się sześć młodszych bitów przesunięcia w kierunku X, a w trzecim to samo dla osi Y. Obrazowo pokazano to w **tab. 2**. Liczby reprezentowane są w kodzie uzupełnieniowym do dwóch. Oznacza to, że między jednym a drugim komunikatem kursor może się przemieścić maksymalnie o +127 lub -128 punktów. Jeśli przemieszczenie będzie większe, program ograniczy je do dopuszczalnego zakresu. Widać to dokładnie na list. 1. Pozycje oznaczone w tab. 2. symbolami LB (Left Button) i RB (Right Button) niosą informację o stanie klawiszy. Jedyńka odpowiada naciśnięciu klawiszowi.

Podczas eksploatacji modelu myszki miałem duże kłopoty z uzyskaniem podwójnego kliknięcia. Początkowo sądziłem, że zbyt duży jest odstęp między jednym a drugim dmuchnięciem. Niestety, ten sam efekt występował przy użyciu zwykłego przycisku podłączonego do złącza Z1. Szybko okazało się, że aby komputer odczytał dwa dmuchnięcia jako jedno podwójne kliknięcie, kursor musi pozostawać przez cały czas idealnie nieruchomy. Mimo że dużo wysiłku włożyłem w wyeliminowanie przypadkowych skoków kursora spowodowanych szu-

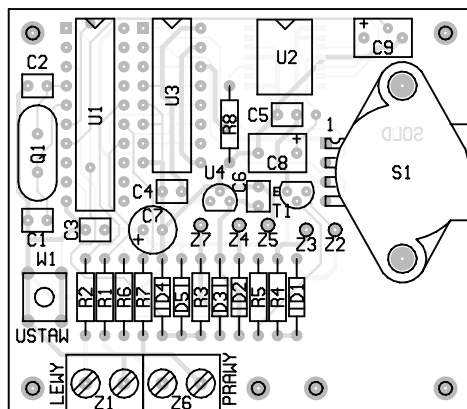
mem na wyjściu akcelerometrów, to ruchy o 1..2 punkty są nieuniknione. Problem podwójnego kliknięcia rozwiązałem w ten sposób, że po każdej zmianie stanu klawiszy ruchy kursora są blokowane na około 0,5 sekundy. Podnosi to znacząco komfort pracy, tylko nieznacznie ją spowalniając.

Montaż

Wszystkie podzespoły myszki montujemy na dwustronnej płytce drukowanej, której układ ścieżek można znaleźć na wkładce wewnątrz numeru, płyty CD-ROM dołączonej do tego numeru EP lub pod adresem www.ep.com.pl/pcb.htm. Rozmieszczenie elementów ilustruje **rys. 5**. Na rysunku nie zaznaczono biegunowości kondensatora elektrolitycznego C6. Dodatni biegun tego kondensatora znajduje się od strony złącza Z5.

Jako złącza: Z2, Z3, Z4, Z5 i Z7 przewidziałem zastosowanie kołków lutowniczych. Nie jest to konieczne, ale ich obecność może później ułatwić ewentualną wymianę uszkodzonego kabla połączeniowego. Montaż proponuję zacząć od ostrożnego wbicia tych pięciu szpilek w płytkę drukowaną.

Następnie montujemy elementy w kolejności od najniższych do najwyższych. Proponuję nie montować na razie rezystora R8, podstawkę pod U3, kondensatora C9 i czujnika S1, gdyż będą one przeszkadzać przy lutowaniu sensora U2. Element ten jest w obudowie do montażu powierzchniowego, dlatego trzeba poświęcić mu trochę więcej uwagi. Oczywiście najlepsza byłaby lutownica na gorące powietrze i pasta lutownicza. W amatorskich warunkach wystarczająco dobre rezultaty można osiągnąć stosując cynę w postaci wielordzeniowego drutu o średnicy 1mm lub mniejszej i lutownicę wyposażoną w ostro zakończony grot. Zwykle zdajemy sobie sprawę, że w pracy z układami scalonymi należy stosować środki ostrożności zapobiegające



Rys.5. Rozmieszczenie elementów na płytce drukowanej.

powstawaniu ładunków elektryczności statycznej. W praktyce różnie z tym bywa i zwykle nic wielkiego się nie dzieje. Chciałbym ostrzec, że ADXL202 są nieco bardziej czułe na elektryczność statyczną niż inne układy. Dlatego nie od rzeczy będzie uziemiona lutownica i opaska uziemiająca na rękę. Lutujemy najpierw jeden pin i dokładnie pozycjonujemy układ. Teraz lutujemy przeciwległe wyprowadzenie i ponownie sprawdzamy, czy końcówki leżą dokładnie na pocynowanych plackach miedzi. Jeśli tak jest, lutujemy pozostałe piny używając przy tym jak najmniejszej ilości cyny.

W czujniku ciśnienia S1 ostrożnie zaginamy wyprowadzenia w odległości około jednego milimetra od obudowy. Końcówka numer jeden jest półkuliście nacięta, dlatego nie powinno być problemów z jej identyfikacją. Przykręcamy obudowę czujnika dwoma śrubkami M3 i dopiero potem lutujemy wyprowadzenia do płytki.

Na koniec montujemy pozostałe brakujące elementy pamiętając, że pod procesor U1 i przetwornik U3 należy zamontować podstawkę. Złącza śrubowe Z1 i Z6 proponuję wlotować w taki sposób, aby kabel wchodził do nich od środka płytki. Dzięki temu, jeśli zdecydujemy się na dołączenie do myszki dodatkowych przycisków, przewody do nich będzie można docisnąć do obwodu drukowanego razem z kablem RS232C. Posłuży do tego blaszka i dwie śrubki M3. Odpowiednie otwory do tego celu znajdują się na płytce. Takie rozwiązanie skutecz-

nie zabezpieczy przewody przed wyrwaniem, a o to w czasie eksploatacji naprawdę nietrudno.

Uruchomienie i kalibracja

Uruchomienie jak zawsze rozpoczynamy od sprawdzenia poprawności montażu. Jeśli nie ma zwarcia między polami lutowniczymi i wszystkie elementy wydają się być obsadzone poprawnie, wyciągamy z podstawki procesor U1 i przetwornik U3. Do tak przygotowanego układu możemy podłączyć zasilanie. Najlepiej do tego celu wykorzystać regulowany zasilacz. Plus zasilacza podłączamy do złącza Z4, a masę do Z5. Włączamy zasilacz i stopniowo zwiększamy napięcie wyjściowe, cały czas sprawdzając napięcie na wyjściu stabilizatora U4. Po przekroczeniu 8V na zasilaczu, napięcie wyjściowe U4 powinno ustabilizować się na poziomie 5V. W tych warunkach pobierany prąd nie powinien przekraczać 2mA. Jeśli wyniki pomiarów są zgodne z oczekiwaniami, wyłączamy zasilanie i wkładamy w podstawki procesor i przetwornik.

Przy wyłączonym komputerze podłączamy naszą myszkę do portu RS232C i włączamy komputer. Pomimo że myszka leży nieruchomo, kursor może wykonywać niewielkie ruchy na ekranie. Jest to normalny objaw i świadczy o poprawnym działaniu. Przy pierwszym włączeniu myszka będzie się zachowywać tak, jakby cały czas miała naciśnięty któryś klawisz. Dlatego nie ruszając jej z miejsca, powinniśmy czym prędzej dokonać kalibracji czujnika ciśnienia. W tym celu upewniamy się, że nikt nie dmucha w rurkę czujnika i naciskamy na chwilę przycisk USTAW (W1). Na czas wciśnięcia przycisku kursor powinien znieruchomieć. Wynik kalibracji jest zapisywany w pamięci nieulotnej, dlatego wystarczy wykonać ją tylko raz. Od tego momentu myszka jest gotowa do pracy.

Konstrukcja mechaniczna

Nie da się ukryć, że myszka w postaci płytki drukowanej z zamontowanymi elementami jest właściwie bezużyteczna. Dlatego

chciałbym przedstawić kilka rozwiązań mechanicznych, które zastosowałem w modelu. Przede wszystkim myszkę należy zamknąć w niewielkiej obudowie z tworzywa sztucznego. Z obudowy powinny wystawać oba króćce czujnika ciśnienia. Standardowo rurkę podłączamy do tego, który znajduje się bliżej płytki drukowanej.

Do stabilnego zamocowania myszki na głowie wykorzystałem zwykłą czapkę „bejsbolówkę“ założoną daszkiem do tyłu. Na górze czapki zostały przyszyte rzepy (dziękuję Aniu!). Drugą część rzepów przykleiłem do spodu obudowy już samodzielnie. W daszku czapki wykonałem kilka otworów. Dwa z nich służą do zamocowania kabla do komputera. Dzięki temu ten dość długi przewód nie ściąga myszki do tyłu przy każdym ruchu. Mniejsza jest też szansa, że myszka wylądnie na podłodze, gdy ktoś zahaczy o kabel.

Pozostałe otwory służą do zamocowania pałaka z dość twardego drutu. Pałak przechodzi koło ucha i kończy się na wysokości ust spłaszczonym oczkiem. W to oczko wciśnięty jest ustnik, który wykonałem ze zbiorniczka „aparatu do iniekcji“, czyli popularnej kroplówki. Podgrzany w gorącej wodzie zbiorniczek spłaszczyłem w połowie wysokości. Po odcięciu górnej pokrywy powstał całkiem zgrabny ustnik. Z drugiej strony zbiorniczka wychodzi wężyk idealnie pasujący do czujnika ciśnienia.

Czujnik ciśnienia jest w dużym stopniu odporny na działanie wilgoci. Mimo to ustnik i rurkę po umyciu należy każdorazowo przedmuchać i dokładnie osuszyć.

Możliwości adaptacji

Konstrukcja i oprogramowanie myszki daje dość duże możliwości adaptacji do indywidualnych potrzeb.

Zacznijmy od czułości. Przez zmianę wartości rezystora R8 możemy wpływać na okres sygnału wyjściowego z akcelerometrów. Przy minimalnej dopuszczalnej rezystancji, wynoszącej 125kΩ, zmniejszamy czułość dziesięciokrotnie. Jakkolwiek największą wartością zalecaną dla aplikacji

wykorzystujących wyjścia PWM jest $1,25M\Omega$, to podłączając rezystor $2M\Omega$ można uzyskać całkiem stabilne zwiększenie czułości.

W celu przystosowania myszki dla osób, u których występuje bezwiedne drżenie głowy, można zwiększyć pojemności kondensatorów C8, C9. Efektem tego będzie wolniejsza reakcja myszki na ruch głowy, przy czym czułość nie ulegnie zmianie. Stosując kondensatory elektrolityczne należy zwrócić uwagę na biegunowość. Odpowiednie symbole znajdują się na płycie drukowanej.

W modelu myszki obsługa klawiszy polega na dmuchaniu lub zasysaniu powietrza z rurki podłączonej do czujnika ciśnienia. Dmuchięcie odpowiada lewemu, a zassanie prawemu klawiszowi. Działanie możemy w prosty sposób odwrócić podłączając rurkę do drugiego króćca czujnika.

Jeszcze innym rozwiązaniem jest podłączenie rurek do obu

króćców. W takim przypadku dmuchnięcie w jedną rurkę odpowiada wciśnięciu lewego klawisza, a dmuchnięcie w drugą jest tożsame z naciśnięciem prawego klawisza. Dla osób z częściowo sprawnymi rękami można rurki przedłużyć i zakończyć gumowymi gruszkami. W ten sposób po ponownym skalibrowaniu czujnika ciśnienia przyciskiem USTAW, otrzymamy wygodne w obsłudze wyłączniki pneumatyczne.

Jeśli komuś wystarczą wyłączniki elektryczne, nie ma potrzeby stosowania „instalacji pneumatycznej“. Można wówczas znacząco obniżyć koszt myszki rezygnując z montowania czujnika S1 i przetwornika U3. W takim przypadku trzeba kawałkiem przewodu zewrzeć na stałe wyprowadzenia mikroprzełącznika W1. W momencie inicjalizacji procesor sprawdza stan tego przycisku i jeśli jest on zwarty, pomija w dalszej

pracy procedury odczytu czujnika ciśnienia.

Tomasz Gumny, AVT
tomasz.gumny@ep.com.pl

"Dziękuję firmie ALFINE z Poznania za udostępnienie układów ADXL202 /Analog Devices/ i firmie UNIPROD-COMPONENTS z Gliwic za próbki układów UTI /Smartec/" - to zdanie napisałem posługując się opisaną w artykule myszką za pomocą programu wirtualnej klawiatury. Zajęło mi to 13 minut i 54 sekundy, co daje średnią szybkość pisania wynoszącą jeden znak na 5,5 sekundy.

Dziękujemy Panu Bogdanowi Janiakowi za pomoc w przygotowaniu zdjęcia na naszą kwietniową okładkę - Redakcja EP.

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/pcb.html> oraz na płycie CD-EP05/2000 w katalogu PCB.