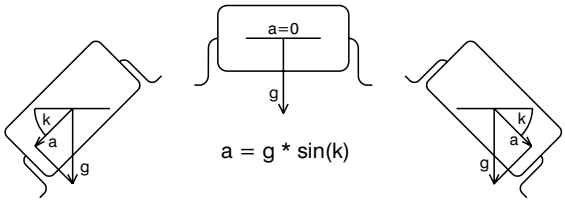
Myszka komputerowa dla osób niepełnosprawnych, część 1

PROJEKT Z OKŁADKI

AVT-862

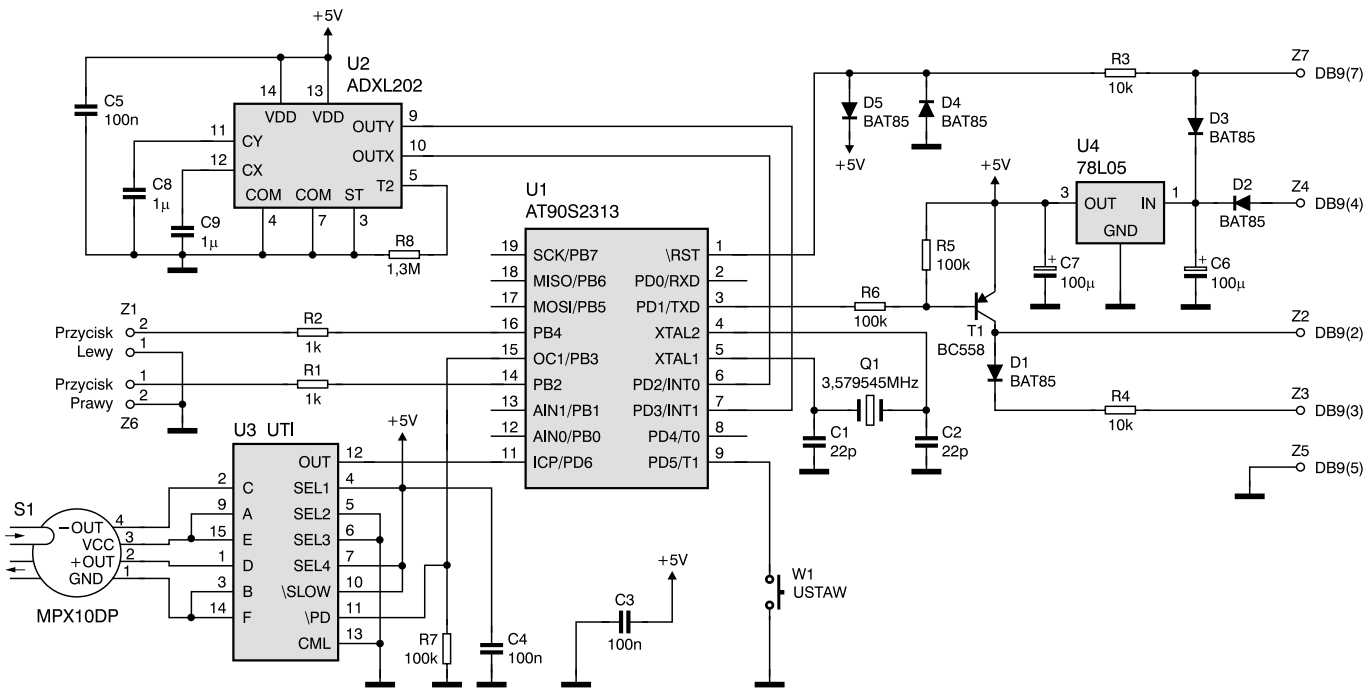
Przedstawione poniżej urządzenie ma dość nietypową konstrukcję. Zawiera dwa nowoczesne akcelerometry i czujnik ciśnienia z precyzyjnym przetwornikiem analogowo-cyfrowym. Sygnały z tych sensorów przetwarza szybki procesor o architekturze RISC. To wszystko służy... przesuwaniu kursora na ekranie komputera.

Czy wykorzystanie tak zaawansowanych technologii jest konieczne do budowy tytułowej myszki komputerowej? Tak, jeśli założymy, że myszka ma umożliwiać dostęp do komputera osobie, która może pracować tylko głową. I to dosłownie. Opisane poniżej urządzenie jest bowiem przeznaczone dla osób z niedowładem rąk i nóg. Zdarza się, że takie schorzenia są wynikiem choroby i z tym musimy się pogodzić. Niestety, często wiąże się to z urazami, które powstają w czasie wypadków komunikacyjnych lub lekkomyślnych zabaw na przykład nad wodą. Wiele, zbyt wiele ofiar tych wypadków, to ludzie młodzi mający przed sobą wiele lat życia i to życia, które wcale nie musi być wegetacją.



Rys. 1. Wpływ przechyłu na wartość przyspieszenia ziemskiego widzianego przez akcelerometr.

Radio, telewizja, telefonia stacjonarna, a ostatnio komórkowa, komputery i Internet - to wszystko ma służyć zaspokajaniu naszych potrzeb. Wiele osiągnięć techniki jest dla nas, ludzi zdrowych, tylko kolejnym udogodnieniem. Dla osób niepełnosprawnych niektóre z tych mediów mogą stać się prawdziwym dobrodziejstwem. Ktoś, kto nie spędził całych miesięcy unieruchomiony w łóżku, gdzie jedyną dostępną rozrywką jest książka lub ogłupiająca telewizja, nie jest w stanie wyobrazić sobie, czym dla osoby sparaliżowanej, może stać się komputer i dostęp do Internetu. Nie, nie tylko źródłem rozrywki. Na internetowych grupach dyskusyjnych taka osoba może rozmawiać. Strony *www* są niewyczerpanym źródłem wiedzy. Dzięki telepracy, osoba niepełnosprawna może



Rys. 2. Schemat elektryczny myszki.

znaleźć zatrudnienie. Nie twierdzę, że to wszystko jest łatwe, ale jestem przekonany, że to jest możliwe.

Do pokonania istniejących barier nie zawsze są potrzebne marmurowe podjazdy dla wózków inwalidzkich. Czasem wystarczą niepozorne rzeczy. Mam nadzieję, że przedstawiona myszka stanie się dla kogoś takim pomocnym drobiazgiem.

Na początek proponuję prosty eksperyment, który uzmysłowi nam, na jakie problemy napotykają osoby z niedowładem rąk. Usiądź sobie wygodnie przed komputerem, dłonie połóż na kolanach. Teraz napisz na komputerze swoje imię. Gdzie te łapy? Z powrotem na kolana! Możesz wziąć w usta ołówek i spróbować za jego pomocą naciskać klawisze. Jeśli nie znajdziesz na biurku ołówka, pozostaje pisanie nosem. To nie jest żart. Gdy już się podpiszesz, spróbuj poruszyć myszką i kliknąć! Ciężko, nieprawdaż? A teraz pomyśl, że i tak jesteś w dobrej sytuacji, bo poruszasz swobodnie tułowiem, czego osoby sparaliżowane zwykle robić nie mogą.

Konstruując urządzenie, staram się myśleć o przyszłym użytkowniku, o jego potrzebach i możliwościach. Wy, drodzy Czytelnicy, budując różnego rodzaju urządzenia zazwyczaj widzicie siebie jako ich przyszłych użytkowników. Nie sądzę, aby wśród Czytelników EP znalazło się wiele osób dotkniętych tetraplegią. Natomiast nie mam wątpliwości, że niektórzy z Was znają osoby, którym myszka mogłaby się przydać.

Ze skruchą muszę przyznać, że przez te kilka lat odkąd mam okazję pisywać dla Elektroniki Praktycznej, nie zaprojektowałem żadnego układu z myślą o osobach niepełnosprawnych. Niewielką pociechą jest fakt, że z czystym sumieniem mógłbym grzmotnąć w pierś prawie całe grono redakcyjne. Oszczędziłbym jedynie Jurka Szczesiula, który stworzył program klawiatury wirtualnej. Program ten zostanie przedstawiony w jednym z kolejnych numerów EP. W skrócie powiem tylko, że dzięki programowi wirtualnej klawiatury można, posługując się jedynie myszką, szybko i wygodnie wpisywać teksty do komputera. A jeśli do komputera będzie podłączona opracowana przeze mnie myszka, to naprawdę wystarczy do tego tylko zdolność do wykonywania niewielkich przechyłów głowy oraz umiejętność dmuchnięcia i zassania powietrza przez ustnik.

Najważniejsze parametry myszki przedstawiono w **tab. 1**. Przyjrzyjmy się im bliżej. Nasza myszka jest widziana przez komputer jako standardowa myszka pracująca w systemie Microsoft Serial Mouse. Dzięki temu nie wymaga specjalnych sterowników i pracuje poprawnie w środowisku graficznym Windows i programach działających w systemie DOS.

Elektronika myszki mieści się w niewielkiej obudowie wielkości paczki papierosów. Do poprawnej pracy konieczne jest umieszczenie myszki na głowie osoby obsługującej. Przechylanie głowy do przodu i do tyłu powoduje na ekranie przemieszczanie kursora na dół i do góry. Analogiczne ruchy głowy w lewo i prawo powodują zmianę pozycji kursora w osi X.

Wyszedłem z założenia, że projektowana myszka powinna umożliwić pracę osobie całkowicie sparaliżowanej, która może poruszać jedynie głową. Dlatego myszka jest zaopatrzona w czujnik pneumatyczny z ustnikiem. Dmuchnięcie w rurkę jest widziane przez komputer jako naciśnięcie lewego klawisza myszy. Zassanie powietrza odpowiada naciśnięciu prawego klawisza.

Niezależnie od tego, do myszki można podłączyć dwa dodatkowe wyłączniki. Dzięki temu do naciskania klawiszy można wykorzystać ruchy np. władnego palca.

Tab. 1. Podstawowe parametry myszki.

- ✓ praca w standardzie MS-MOUSE;
- ✓ zmiana pozycji kursora przez przechylanie głowy;
- ✓ przełącznik pneumatyczny:
  - ✗ dmuchnięcie - klawisz lewy;
  - ✗ zassanie - klawisz prawy;
- ✓ możliwość podłączenia dodatkowych wyłączników;
- ✓ zasilanie z portu szeregowego;
- ✓ pobór prądu około 6mA.

Możliwości adaptacji jest wiele i kilka propozycji zostanie przedstawionych na końcu artykułu.

Niewielki pobór prądu umożliwia zasilanie myszki z portu szeregowego. Takie rozwiązanie jest wygodne i powszechnie stosowane.

Założenia projektu

Jak zwykle w takich przypadkach, proces konstruowania rozpocząłem od przeglądu urządzeń oferowanych na rynku i spełniających podobne funkcje. Mimo dość intensywnych poszukiwań nie udało mi się znaleźć zbyt wielu gotowych rozwiązań.

W jednym z najciekawszych, jakie znalazłem, zastosowano reflektor oświetlający twarz osoby siedzącej przed monitorem światłem z zakresu podczerwieni. Obraz z umieszczonej obok miniaturowej kamery był przeszukiwany pod kątem najjaśniejszego punktu. Ruchy tego punktu były przenoszone na kursor. Wielką zaletą takiego rozwiązania jest to, że operator nie jest w żaden sposób „uwiązany“ do komputera. Wystarczy umieścić gdzieś na twarzy niewielką, odblaskową naklejkę. Więcej szczegółów na temat tego rozwiązania można znaleźć na internetowej stronie *www.orin.-com/access/headmouse*.

Pozostałe rozwiązania bazowały na miniaturowych żyroskopach i bardziej przypominały hełmy do oglądania wirtualnej rzeczywistości, niż proste urządzenia wskazujące. Wyjątkiem była bezprzewodowa gyro-mouse, którą można obejrzeć na stronie *www.advancedperipheral.com/gyro.htm*.

Cechą wspólną gotowych rozwiązań jest nieproporcjonalnie wysoka cena, częstokroć przekraczająca cenę dobrej klasy komputera.

Początkowo założyłem, że ruch kursora w poziomie będzie wymuszany przez kręcenie głowy w lewo i w prawo. Pochylanie głowy do tyłu i do przodu miało przesuwać kursor odpowiednio w górę i w dół.

W pierwszym odruchu zamierzałem skonstruować myszkę w oparciu o żyroskopy. Znalazłem nawet miniaturowe żyroskopy elektromagnetyczne, które do działania wykorzystują efekt Coriolisa. Firma Gyration (*www.gyration.com*) zmieściła dwa takie żyroskopy w obudowie o wymiarach 25x25x20mm. Byłem zdecydowany zastosować te ciekawe elementy, niestety, nie do pokonania okazała się bariera cenowa. Dodatkowo zniechęcała mnie konieczność stosowania specjalizowanych układów do odczytywania sygnałów wyjściowych.

Potem wymyśliłem, żeby do mierzenia pochylenia głowy w kierunku przód-tył wykorzystać czujnik przyspieszenia pracujący jako poziomnica (patrz EP9/99). Teraz pozostało już tylko wykrywanie kręcenia głową. Potencjalnie nadawałby się do tego czujnik pola magnetycznego. Wymaga on jednak oddalenia od źródeł zakłóceń elektromagnetycznych i precyzyjnego wypoziomowania. Okolice komputera trudno nazwać obszarem o niskim poziomie zakłóceń, a poziomowanie trudno pogodzić z koniecznością pochylania głowy.

W sumie musiałem zrezygnować z wykrywania kręcenia głową, które wydawało się bardziej naturalne, na rzecz przechylania, które jest łatwiejsze do mierzenia. Dzięki przyjęciu takiego rozwiązania, mogłem w obu płaszczyznach zastosować stosunkowo tanie akcelerometry.

Mój wybór padł na układ ADXL202 firmy Analog Devices. W ceramicznej, 14-nóżkowej obudowie do montażu powierzchniowego, znajdują się dwa ustawione prostopadle akcelerometry. Układ wymaga do pracy niewielu elementów zewnętrznych, a dzięki wyjściom PWM doskonale nadaje się do współpracy z mikrokontrolerem. Dodatkową zaletą jest standardowe zasilanie (4,75..5,25V) i niewielki pobór prądu (<1mA). Jednym słowem wymarzony element do naszej myszki.

Akcelerometry zawarte w układzie ADXL202 mogą mierzyć przyspieszenia w zakresie ±2g. Czułość przetwarzania jest stała i wynosi 12,5%/g z tolerancją ±2,5%. Oznacza to, że współczynnik wypełnienia prostokątnego przebiegu wyjściowego zmienia się o około 12,5% przy zmianie przyspieszenia o 9,81m/s^2.

Przy idealnie poziomym ustawieniu akcelerometru, przyspieszenie ziemskie przez niego mierzone wynosi zero. Przy odchyleniu przyrządu od poziomu o określony kąt, mierzona wartość przyspieszenia będzie rosnąć zgodnie z wartością sinusa kąta odchylenia. Ilustruje to **rys. 1**. Dla 90 stopni sinus osiąga wartość jeden i w takim przypadku mierzone przyspieszenie wyniesie 1g.

Teraz wyobraźmy sobie akcelerometr leżący na głowie. Gdy trzymamy głowę prosto, akcelerometr leży w przybliżeniu poziomo. Przechylając głowę powodujemy odchylenie czujnika od poziomu w jedną lub drugą stronę. Bez większych trudności można przechylić głowę na boki o 45 stopni, a w przód i w tył nawet więcej. Tyle tylko, że przy tak dużych odchyleniach trudno obserwować ekran monitora. Dlatego niezbędne przechylenie głowy powinno być mniejsze od 30 stopni w każdym kierunku.

Uwzględniając teraz fakt, że odchylenie następuje w obie strony, mierzona wartość przyspieszenia będzie się zmieniać od -0,5g do +0,5g. Sumaryczna zmiana przyspieszenia o 1g skutkuje zmianą współczynnika wypełnienia sygnału wyjściowego o 12,5%. Przy okresie sygnału wyjściowego ustalonym na 10,4ms, moment zmiany stanu na wyjściu PWM przesunie się o 1,3ms.

Wiemy już, o ile zmieni się czas impulsu wyjściowego. Przyszła pora na określenie, ilu punktom na ekranie powinno to odpowiadać. Rozdzielczość standardowych kart graficznych wynosi obecnie od 640x480 do 1024x768 punktów. Myszka powinna mieć jednak nieco większą czułość. Już wyjaśniam dlaczego.

Przypuśćmy, że podchodzimy do komputera. Pierwsze co robimy, to szukamy kursora na ekranie. Okazuje się, że jest przy

prawej krawędzi ekranu. Przesuwamy myszkę w lewo. Gdy podkładka się skończy, podnosimy myszkę do góry, przenosimy w prawo, kładziemy na podkładce i znowu przesuwamy w lewo. Te czynności powtarzamy tak długo, aż kursor znajdzie się tam, gdzie powinien.

A teraz wyobraźmy sobie, że nie możemy odrywać myszki od podłoża i mamy ograniczone pole do jej przesuwania. Takie warunki dość dobrze ilustrują sytuację przy posługiwaniu się myszką nagłowną. Załóżmy, że kursor znowu jest po prawej stronie ekranu, a myszka jak na złość z lewej strony pola. Co robimy? Przesuwamy myszkę w prawo. Wkrótce kursor, nie mogąc wyjechać dalej „oprze“ się o prawą krawędź ekranu. Przesuwamy myszkę jeszcze trochę, aż dotrzemy do prawej krawędzi podkładki. Teraz wracamy myszką na środek pola. Taki sam ruch wykona kursor ustawiając się na środku ekranu. I o to nam chodziło.

Aby jednak coś takiego było możliwe, konieczny jest pewien zapas rozdzielczości. Wstępnie założyłem, że czułość myszki powinna wynosić około 1300 punktów przy pełnym zakresie odchylenia głowy.

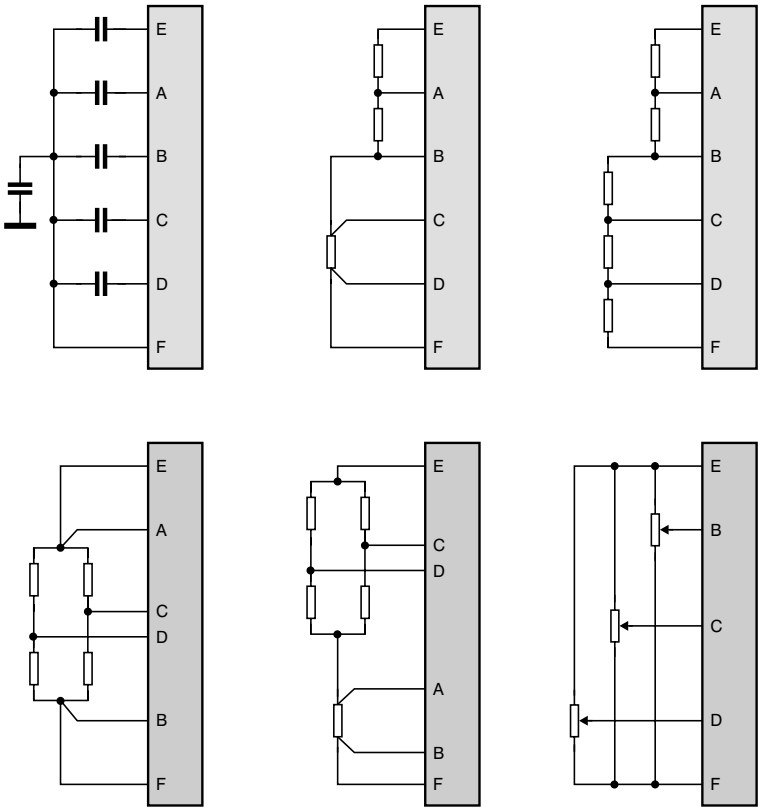
Przeanalizujmy teraz, jaki to ma wpływ na konstrukcję myszki. Przebieg PWM najłatwiej zdekodować zliczając za pomocą wewnętrznych timerów mikrokontrolera czas trwania stanu niskiego i wysokiego. Ustaliliśmy już, że czas trwania tych stanów zmieni się o 1,3ms. Dla uzyskania założonej czułości wewnętrzny timer procesora powinien w tym czasie zliczyć przynajmniej 1300 impulsów. Wynika z tego, że częstotliwość taktująca timer powinna wynosić około 1MHz. W procesorach rodziny MCS-51, a początkowo chciałem zastosować AT89C2051, najwyższa częstotliwość taktująca wynosi 1/12 częstotliwości kwarcu. Oznacza to konieczność taktowania procesora przebiegiem o częstotliwości 12MHz. W takich warunkach „mały atmelek“ pobiera około 10mA, a to już za dużo jak na układ, który ma być zasilany z portu szeregowego komputera.

To zadecydowało o wyborze procesora AT90S2313 z rodziny AVR firmy Atmel. Przebieg taktujący wewnętrzne liczniki w tym procesorze może mieć częstotliwość równą częstotliwości rezonatora kwarcowego, czyli wystarczy taktowanie procesora kwarcem o częstotliwości 1MHz. Dlaczego ta częstotliwość ostatecznie wzrosła do 3,58MHz, wyjaśnię później.

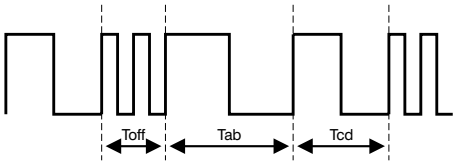
Gdy już miałem akcelerometry i procesor, zabrałem się za poszukiwanie przełączników ciśnieniowych. W tym momencie natrafiłem na nadspodziewanie wiele problemów. Z dmuchania w manometr ciśnieniomierza lekarskiego wiedziałem już, że potrzebuję przełączników o czułości rzędu 15mmHg (2kPa). Okazało się, że łatwiej znaleźć wyłączniki na zakres megapascali. W dodatku potrzebowałem przełączników różnicowych, w których zamknięcie zestyku następuje pod wpływem nadciśnienia (dmuchnięcie) lub podciśnienia (zassanie). Najbliższe spełnieniu tych warunków były niewielkie hydrostaty stosowane w sprzęcie AGD, np. w pralkach i ekspresach do kawy. Tu jednak trudno znaleźć jakiś standardowy typ. W dodatku elementy te są zazwyczaj duże, bo przeznaczone do pracy przy znacznym obciążeniu i pod napięciem 220V.

Ostatecznie zdecydowałem się na zastosowanie czujnika ciśnienia MPX10DP firmy Motorola. Daje on wprawdzie sygnał wyjściowy na poziomie pojedynczych miliwoltów, jest nieskompensowany napięciowo i termicznie, za to jest najtańszy spośród całej rodziny czujników i nadaje się do mierzenia nad- i podciśnienia. Dodatkowo producent wypełnia wnętrze czujnika żelem silikonowym, dzięki czemu jest on w dużym stopniu odporny na wilgoć zawartą w powietrzu.

Koniecznym uzupełnieniem tego czujnika jest przetwornik analogowo-cyfrowy. W tej roli zastosowałem bardzo ciekawy układ UTI firmy Smartec. Nietypowe oznaczenie układu wynika chyba z faktu, że Smartec produkuje tak mały asortyment układów, że nie ma potrzeby ich numerować. Są wśród nich dość popularne czujniki temperatury typu SMT160-30 i mniej znane czujniki wilgotności względnej SMT-RH05.



Rys. 3. Sposoby podłączania różnych czujników do UTI.



Rys. 4. Przebieg na wyjściu układu UTI.

Opis układu

Schemat elektryczny myszki znajduje się na **rys. 2**. Można na nim wyróżnić pięć bloków:
- czujnik przechyłu (U2),
- czujnik dmuchnięcia (S1) z przetwornikiem (U3),
- procesor sterujący (U1),
- interfejs we/wy (T1),
- zasilacz stabilizowany (U4).

Czujnik przechyłu zrealizowałem na układzie ADXL202. Jak wcześniej wspomniałem, układ ten zawiera dwa akcelerometry obrócone o kąt 90 stopni. Do poprawnej pracy układ potrzebuje dwóch kondensatorów filtrujących C8 i C9 oraz rezystora ustalającego R8.

Kondensatory filtrujące determinują czas odpowiedzi czujników przyspieszenia. C8 filtruje sygnał akcelerometru w jednej osi, podczas gdy C9 robi to samo w drugiej osi. Minimalna pojemność tych kondensatorów wynosi 1nF. Przy takiej pojemności akcelerometry mogą wykrywać drgania o częstotliwości 5kHz. W naszej aplikacji wystarczy czas reakcji rzędu dziesiątych części sekundy. Trudno bowiem sobie wyobrazić, aby ktoś mógł poruszać głową szybciej. Ograniczając pasmo przenoszenia czujników do 5Hz, zmniejszamy poziom szumów w sygnale wyjściowym. Przy wartości 1µF wartość szczytowa szumów nie powinna przekroczyć 8mg (g - przyspieszenie ziemskie). Ma to bardzo istotne znaczenie dla stabilności położenia kursora na ekranie. Przy zmianie mierzonego przyspieszenia w granicach 1g i rozdzielczości rzędu 1000 punktów, jeden piksel odpowiada 1mg. Poziom szumów na poziomie 8mg oznacza nieprzewidywalne ruchy kursora o 8 pikseli. Dalsze zwiększanie pojemności kondensatorów C8 i C9 nie powoduje już znaczącego zmniejszenia szumów. Lepsze efekty można uzyskać na drodze programowej, ale o tym pomówimy w części poświęconej oprogramowaniu.

Rezystor R8 ustala okres sygnału wyjściowego w obu kanałach. Przy wartości 1,3MΩ okres sygnału wyjściowego wynosi około 10,4ms. Wyjścia akcelerometrów 9 i 10 (U2) są połączone bezpośrednio z wejściami INT0 (6) i INT1 (7) procesora (U1).

Drugi blok to czujnik ciśnienia S1 z przetwornikiem analogowo-cyfrowym U3. Konstrukcja sensora opiera się na klasycznym mostku rezystancyjnym o stopniu niezrównoważenia zależnym od przyłożonego ciśnienia. Czujnik jest wyposażony w dwa króćce, z których pierwszy doprowadza medium do komory z jednej strony membrany czujnikowej, a drugi do komory z drugiej strony membrany.

Rezystancja mostka widziana od strony zasilania wynosi około 500Ω. Przy zasilaniu z +5V czujnik pobierałby 10mA, a to stanowczo za dużo. Konieczne było zatem znalezienie sposobu na obniżenie pobieranej mocy. Można to zrobić przez obniżenie napięcia zasilającego mostek. Wadą takiego rozwiązania jest zmniejszanie i tak już małego sygnału wyjściowego czujnika. Tu ujawniły się kolejne zalety zastosowanego przetwornika UTI. Ustawienie wejścia \PD (11, U3) w stan niski powoduje nie tylko uśpienie przetwornika i zmniejszenie pobieranego przezeń prądu, ale również wyłączenie zasilania sensora! Usypianiem przetwornika steruje mikrokontroler za pomocą wyjścia PB3 (15, U1) Rezystor R7 wymusza niski stan na tej linii zaraz po pojawieniu się napięcia zasilającego i przez cały czas trwania impulsu zerującego procesor.

Czujnik S1 jest zasilany z układu UTI (U3) napięciem o przebiegu prostokątnym występującym na wyjściach E-F. Rzeczywista wartość napięcia zasilającego mostek jest mierzona na wejściach A-B. Takie rozwiązanie umożliwia prowadzenie pomiarów linią czteroprzewodową i znaczne oddalenie sensora od przetwornika. Właściwe napięcie niezrównoważenia wchodzi na wejścia C-D.

Wyjście przetwornika jest podłączone do pinu PD6 (11, U1).

Nazwa UTI jest skrótem od *Universal Transducer Interface* i muszę przyznać, że nazwa nie jest na wyrost. Do tego przetwornika można podłączać najróżniejsze czujniki: pojemnościowe, rezystory platynowe, termistory, mostki rezystancyjne lub po prostu potencjometry. Przykładowe sposoby podłączenia czujników przedstawiono na **rys. 3**. Układ posiada jedno wyjście, na którym pojawia się kilkufazowy przebieg umożliwiający kontrolerowi autokalibrację i pomiar. Dzięki temu rozdzielczość i liniowość pomiaru jest na poziomie 12 bitów.

W naszej myszce układ UTI pracuje w trybie pomiaru mostka rezystancyjnego o niezrównoważeniu mniejszym niż ±4%. Czas pomiaru wynosi około 12ms. Przy takich ustawieniach na wyjściu pojawia się trójfazowy przebieg (**rys. 4**), w którym czas pierwszej fazy *Toff* umożliwia pomiar offsetu toru pomiarowego, czas drugiej fazy *Tab* określa wartość napięcia zasilającego mostek pomiarowy, a czas trzeciej fazy *Tcd* odpowiada napięciu wyjściowemu mostka. Znając te trzy czasy, można precyzyjnie obliczyć stopień niezrównoważenia mostka.

Sercem myszki, i to szybko bijącym, jest mikrokontroler AT90S2313 taktowany sygnałem o częstotliwości 3,58MHz. Częstotliwość przebiegu zegarowego jest ustalana rezonatorem kwarcowym Q1 z towarzyszącymi kondensatorami C1 i C2. Ten typ procesora posiada 2KB pamięci programu, 128 bajtów pamięci RAM i tyle samo pamięci EEPROM.

Oprócz wspomnianych wcześniej sygnałów z akcelerometrów i przetwornika UTI, do procesora wchodzą jeszcze dwie linie sygnałowe. Zwarcie przewodów na złączu Z1 jest równoznaczne z naciśnięciem lewego klawisza myszki. Aby zasymulować naciśnięcie prawego klawisza, należy zewrzeć wyprowadzenia złącza Z6. Wejścia PB4 (16) i PB2 (14), którymi procesor (U1) czyta stan tych linii, są programowo ustawione jako wejścia z wewnętrznym podciągnięciem do plusa zasilania. Rezystory R1 i R2 zabezpieczają w pewnym stopniu te wejścia przy nie do końca przemyślanych eksperymentach.

Naciśnięcie wyłącznika W1 - „USTAW“ zwiera do masy wejście PD5 (9, U1). Przycisk ten służy do

zerowania czujnika ciśnienia. Kiedy i po co należy go nacisnąć, dowiemy się w części poświęconej uruchamianiu myszki.

Komunikację z komputerem zapewniają myszce dwie linie. Komputer, wystawiając na chwilę (około 100ms) jedynkę logiczną (-12V) na wyjściu RTS (Z7), żąda od myszki nadania komunikatu identyfikującego. Sygnał ten przez rezystor R3 trafia na diody D4, D5 ograniczające jego amplitudę. Tak ukształtowany sygnał wchodzi na wejście zerujące procesora \RST (1, U1). Jeśli teraz komputer „zdejmie“ jedynkę i na wyjściu pojawi się napięcie +12V (oznaczające zero), mikroprocesor rozpocznie wykonywanie zaszytego programu, a tam jedną z pierwszych czynności jest wysłanie identyfikatora do komputera.

Transmisja w drugą stronę wychodzi z procesora nóżką TXD (3, U1) i przez dzielnik złożony z rezystorów R6 i R5 wchodzi na bazę tranzystora T1. Tranzystor ten pełni rolę konwertera poziomów. Gdy na wyjściu TXD procesora jest niski stan logiczny, tranzystor przewodzi i na jego kolektorze występuje napięcie bliskie napięciu zasilania +5V. Kolektor T1 jest połączony za pośrednictwem złącza Z2 z wejściem RD portu RS232C w komputerze, zatem komputer odczyta taki stan jako logiczne zero. Jeśli procesor U1 „wystawi“ na pinie TXD jedynkę, tranzystor będzie zatkany i kolektor, a wraz z nim RD portu RS232C, będzie na potencjale -12V. To ujemne napięcie przychodzi na złącze Z3 z nieaktywnego w tym momencie wyjścia TD portu RS232C. Rezystor R4 ogranicza prąd pobierany z tego wyjścia, a dioda D1 przepuszcza tylko napięcie o ujemnej polaryzacji.

Ostatnim blokiem myszki jest stabilizator napięcia zasilającego. Źródłem zasilania dla myszki są dwie linie portu szeregowego: RTS (Z7) i DTR (Z4). Po uruchomieniu w komputerze programowego sterownika myszki, na obu liniach występuje niski stan logiczny, co odpowiada napięciu około +12V. Prąd z tych linii po przejściu przez diody D2 i D3 wchodzi na wejście stabilizatora U4. Wyjścia portu szeregowego charakteryzują się stosunkowo dużą rezystancją wyjściową, dlatego nie ma potrzeby stosowania rezystorów wyrównujących prąd pobierany z obu linii.

Podane wcześniej napięcia na wyjściach portu RS232C są wartościami nominalnymi. Już przy niewielkim obciążeniu napięcie spada do około ±10V. W komputerach, które sprawdzałem, przy obciążeniu rzędu 8mA napięcie nie spadało poniżej ±8V. Mniej więcej takiego napięcia potrzebuje na wejściu stabilizator U4 typu 78L05 do poprawnej pracy. Wprawdzie po drodze są jeszcze diody D2 i D3, ale dzięki zastosowaniu diod Schottky'ego spadek napięcia na nich jest pomijalnie mały (0,5V).

Znacznie korzystniej byłoby zastosować stabilizator typu „low-drop“, któremu do poprawnej pracy wystarczy, że napięcie zasilające będzie o kilkaset miliwoltów wyższe od napięcia wyjściowego. Pierwotnie planowałem zastosowanie w tym miejscu stabilizatora typu LM2931-5.0. Dość często można spotkać się z opinią, że układy te są mało stabilne i potrafią się wzbudzać. Skutkiem tego jest przegrzewanie elementu nawet przy teoretycznie niewielkiej wydzielanej mocy. Zamierzałem rozprawić się z tą opinią, bo nigdy dotąd nie miałem z nimi problemów. Ten typ stabilizatora wymaga tylko nieco staranniejszego blokowania wyjścia. Powinno używać się do tego celu kondensatorów aluminiowych o małej wartości ESR lub kondensatorów tantalowych. Ku memu rozczarowaniu okazało się, że układ LM2931 jest dodatkowo uczulony na dużą rezystancję źródła zasilającego. Na nic się zdało blokowanie wyjścia i wejścia. Ostatecznie musiałem przeprosić stary poczciwy stabilizator 78L05.

Wejście stabilizatora oraz zasilanie wszystkich układów scalonych zablokowano kondensatorami C6, C5, C4 i C3.

Wyjście stabilizatora zostało zablokowane kondensatorem elektrolitycznym C7. Poprawia on odpowiedź stabilizatora na gwałtowne zmiany obciążenia. Musimy pamiętać, że myszka pobiera prąd impulsowo, a chwilowy pobór prądu może wynosić kilkanaście miliamperów. Pojemność tego kondensatora nie powinna być mniejsza od 100µF.

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-EP04/2000 w katalogu* ***PCB****.*

WYKAZ ELEMENTÓW

Rezystory
R1, R2: 1kΩ/0,25W
R3, R4: 10kΩ/0,25W
R5, R6, R7: 100kΩ/0,25W
R8: 1,3MΩ/0,25W

Kondensatory
C1, C2: 22pF
C3, C4, C5: 100nF/63V
C6, C7: 100µF/16V
C8, C9: 1µF/63V

Półprzewodniki
D1, D2, D3, D4, D5: BAT85
S1: MPX10DP
T1: BC558
U1: AT90S2313-10PC (zaprogramowany)
U2: ADXL202JQC
U3: UTI
U4: 78L05

Różne
Q1: rezonator kwarcowy 3,579545MHz
Z1, Z6: ARK2 do druku
Z2, Z3, Z4, Z5, Z7, Z9: kołki lutownicze
podstawka pod układ scalony DIL20
podstawka pod układ scalony DIL16
mikroprzełącznik do druku
Gniazdo D-SUB 9pin z obudową
Kabel 5-żyłowy o długości 3m