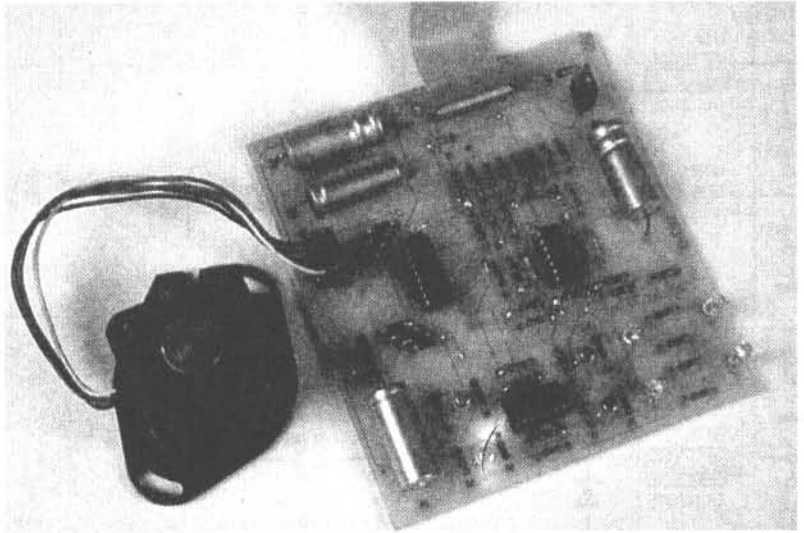


Dwukierunkowy interfejs do silnika krokowego

Silniki krokowe są używane do precyzyjnego ustalania pozycji i odwrotnie mogą być także stosowane jako detektory przesunięcia. Silniki takie są na przykład stosowane do przemieszczania głowic w napędach dyskietań a także w drukarkach i ploterach. Były one dotychczas stosunkowo drogie, ostatnio jednak staniały, zwłaszcza silniki napędów dyskietań o pełnej wysokości, na które zapotrzebowanie znacznie spadło. Opisywany interfejs umożliwia użytkowanie silnika w dwóch trybach - silnika i detektora przesunięcia.



Działanie silników krokowych

Zasadę działania silnika krokowego łatwo zrozumieć przy pomocy rys. 1, na którym wirnik pokazano w postaci namagnesowanego pręta a stojan jako wiele ułożonych wokół niego elektromagnesów. Wiedząc, że przyciągają się przeciwne bieguny, jasnym się staje, że kolejne wzbudzenie cewek elektromagnesów B1, B2, B3, ... spowoduje obracanie się wirnika w kierunku zgodnym z kolejnością wzbudzenia cewek i zatrzymanie się przed ostatnią zasiloną cewką. Przepływ prądu przez uzwojenie w tego typu silnikach zwykle pozostawia się po jego zatrzymaniu dla podtrzymania pozycji.

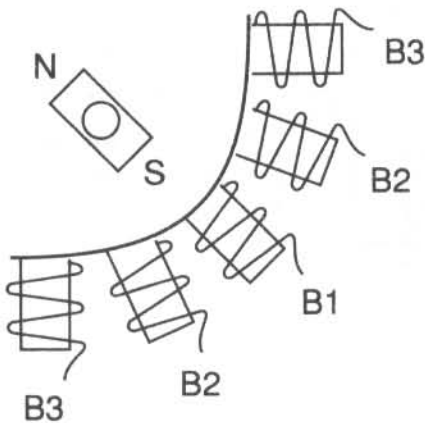
Schemat na rys. 1 jest symboliczny dla lepszej czytelności, w rzeczywistości bowiem pole magnetyczne jest ukształtowane odpowiednim wykresem wirnika.

Osiągi silników

Najważniejszym parametrem silnika krokowego jest wynikająca z konstrukcji precyzja jego ustawiania. Kąt obrotu jest równy ilości kroków pomnożonej przez ilość stopni na krok. Zwykle silniki te mają od 100 do 200 kroków na obrót.

Gdy uruchamia się silnik szybko przesuwać pole, w miarę wzrostu szybkości maleje moment (cewki silnika stawiają opór zmianom prądu). Poczynając od pewnej szybkości wirnik przestaje się kręcić, a silnik wibruje. Nie jest to jednak na szczęście dla niego szkodliwe.

W przypadku pobudzenia dwóch cewek równocześnie, magnes ustawi się pomiędzy nimi stwarzając możliwość przesuwania wirnika co pół kroku. Jeżeli natomiast prąd będzie szybko przełączany na przemian pomiędzy tymi cewkami, ale w stosunku zmiennym (czas przepływu



Rys. 1. Ilustracja pogłądowa wycinka silnika krokowego

wu prądu przez dwie cewki nie jest jednakowy), za cenę skomplikowania układu sterującego można podzielić krok na jeszcze mniejsze części (jest to tzw. multi step.)

Dla zachowania pełnej precyzji pozycjonowania podczas pracy w trybie „przeskakiwanie kroków“, musi być zapewniona dostateczna kontrola szybkości rozruchu i zatrzymywania silnika, tym łagodniejsza, im bardziej jest on obciążony.

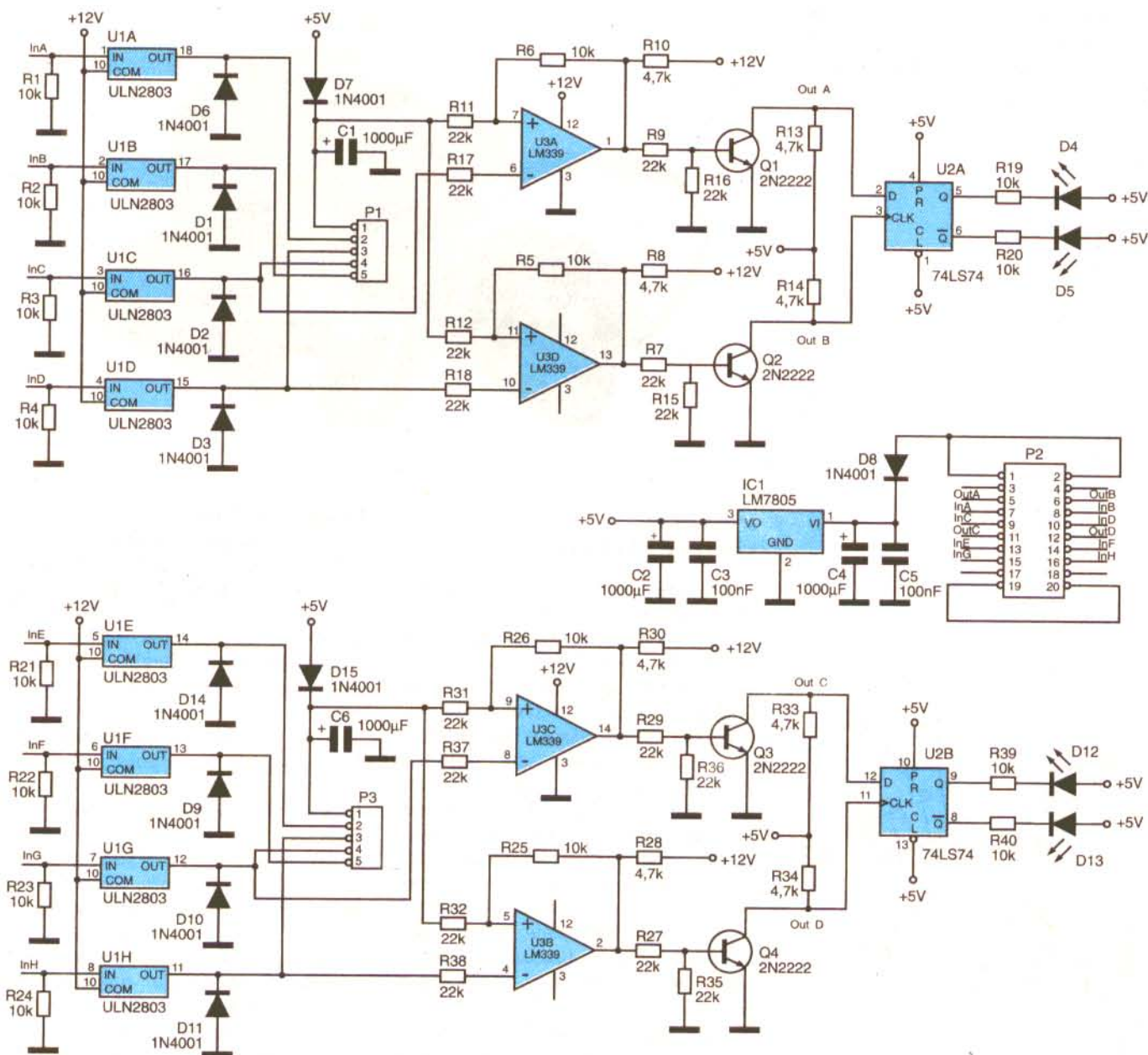
Rodzaje silników

Istnieje wiele rodzajów silników krokowych, zależnie od elektrycznej

konfiguracji uzwojeń. Tanie silniki, stosowane do napędów dyskietek, mają cztery uzwojenia połączone ze sobą, a więc pięć wyprowadzeń. Ich wspólny punkt można łatwo zidentyfikować za pomocą omomierza, oporność pomiędzy nim a pozostałymi czterema wyprowadzeniami jest jednakowa. Oporność ta jest niska (75Ω) i przez uzwojenie może stałe płynąć prąd 150mA. Wspólny punkt uzwojeń łączy się z dodatnim biegunem zasilania, a uzwojenia są kolejno łączone z masą za pośrednictwem tranzystorów. Zasto-

sowania takich właśnie silników krokowych dotyczy ten artykuł.

W napędach dysków twardych stosuje się wydajniejsze silniki, tzw. bipolarne, o czterech wyprowadzeniach, czyli dwóch uzwojeniach. Do ich poprawnego działania trzeba odwracać kierunek prądu w uzwojeniach za pomocą układów mostkowych. W tym zastosowaniu wymagane są jak najmniejsze wymiary i ciężar oraz jak największa szybkość. Wadą tego rodzaju silników jest większy stopień skomplikowania układu sterującego. Istnieje też



Rys. 2. Schemat elektryczny interfejsu do sterowania silnika krokowego

wersja silników o sześciu wyprowadzeniach, które mogą działać zarówno w układzie bipolarnym jak i unipolarnym.

Detekcja przesunięcia

Jeżeli pojawienie się pola elektrycznego, a więc prądu, wywołuje ruch, to na zasadzie odwrotności ruch powinien wywołać pojawienie się prądu (lub napięcia). Charakterystyki silników krokowych do takiego zastosowania nie są podawane przez producentów, jednak po wykonaniu licznych prób okazało się, że detekcja ruchu jest możliwa, ale z pewnym ograniczeniem. Opisany tu prosty układ elektroniczny umożliwia detekcję przesunięcia jedynie co drugi krok, ale w praktyce zupełnie to nie przeszkadza.

Opisany interfejs pozwala sterować dwoma silnikami.

Schemat elektryczny

Schemat jest przedstawiony na rys. 2. Zastosowane zostały silniki o pięciu wyprowadzeniach, czyli czterech uzwojeniach.

Łatwo rozróżnić część detekcji przesunięcia i część sterowania silnikiem. Wspólny punkt uzwojeń jest połączony z +5V; jest to niewielkie napięcie, ale zupełnie wystarczające do obracania kółka napędowego.

Do sterowania silnikami zastosowano układ ULN2803. Zawiera on osiem tranzystorów w układzie Darlingtona oraz diody zabezpieczające przed przepięciami w uzwojeniach. Jeden taki układ wystarcza do sterowania dwoma silnikami. Jeżeli chce się wprawić silnik w ruch, trzeba zasilić jego uzwojenia w kolejności właściwej dla pożądanego kierunku obrotów. Gdy wirnik znajdzie się w zamierzonej pozycji, należy wyłączyć zasilanie uzwojeń dla umożliwienia ręcznej regulacji. Można również pozostawić włączone zasilanie, aby zablokować ręczną regulację. Po wyłączeniu zasilania uzwojeń, każdy obrót wirnika powoduje pojawienie się na nich napięcia.

Druga część schematu składa się z komparatorów połączonych w przerzutniki Schmitta. Ich ujemne wejścia są połączone przez uzwojenia silnika z +5V, dodatnie zaś pomiędzy +5V a wyjście. Jeżeli napięcie wyjściowe wynosi +12V, napięcie na wejściu dodatnim wy-

nosi +5 + E, rezystory wejścia dodatniego tworzą bowiem sumator. Stan ten jest stabilny (w rzeczywistości napięcie to wynosi +4,4 + E ze względu na diodę D7).

Jeżeli kółko napędowe zostanie pokręcone, w uzwojeniu zostanie wzbudzony dodatni impuls, który wprowadzi do ujemnego wejścia komparatora napięcie dodatnie, wyższe od napięcia na wejściu dodatnim, co wywoła przerzut. Napięcie wyjściowe komparatora spadnie do zera, a napięcie na jego wejściu dodatnim spadnie do +5/P (gdzie P jest zbliżone do E). Napięcie na wejściu dodatnim pozostaje wciąż +5V, stan ten jest więc stabilny. Detekcja została dokonana.

Przy dwóch detektorach, za pomocą prostego przerzutnika D daje się określić kierunek obrotu silnika. Kierunek ten jest sygnalizowany świeceniem LED.

Sygnaly logiczne zostały doprowadzone do złącza, do którego można przyłączyć przewód taśmowy. Procesor sterujący nie został wmontowany na płytce. Czytelnicy mogą posłużyć się układami 68075P3 (program sterujący został podany w dalszej części artykułu), a ci, dla których assembler nie jest drugim językiem, chipem zawierającym interpreter Basica. Zresztą do pierwszych prób można wykonać kabel Centronics, a w licznych artykułach można znaleźć programy sterujące tym interfejsem napisane w Basicu lub w C.

Zmiana trybu działania

Z tym są pewne problemy. Po każdej zmianie trybu działania (detekcja/napęd) nie wiadomo, w jakiej pozycji względem uzwojeń znajduje się wirnik. W praktyce wszystko zależy od programu. Trzeba po prostu założyć, że oprogramowanie detekcji przesunięcia nie uaktualnia parametrów sterowania silnikiem, a ponadto po zatrzymaniu napędzanego silnika komparatory pozostają w przypadkowym stanie. W najgorszym przypadku sprowadza się to do obrotu silnika o krok w kierunku przeciwnym do zamierzonego i analogicznie, za pierwszym razem może się zdarzyć odwrotna detekcja kierunku.

Błąd pierwszego kroku jest błędem bardzo małym, jeżeli wziąć pod uwagę przewidywane zastosowania, trzeba jednak podkreślić, że

błąd ten zdarza się tylko przy zmianie trybu działania.

Sprzęg myszy

Należy zaznaczyć, że niektóre komputery, takie jak ATARI i pierwsze Macintosze, posługują się sprzęgiem myszy, który produkuje takie same sygnały, jak wysyłane z wyjść komparatorów. Komputery te można sterować tym rodzajem układów. (Nowe MAC'ci mają wyjście myszy razem z klawiaturą!)

W przypadku PC sygnał ten jest przetwarzany przez mikrosterownik myszy w kod dwójkowy, następnie informacja ta zostaje przesłana łączem szeregowym do komputera (zazwyczaj dla tych, którzy nie chcą blokować myszą łącza szeregowego, pozostają karty myszy...).

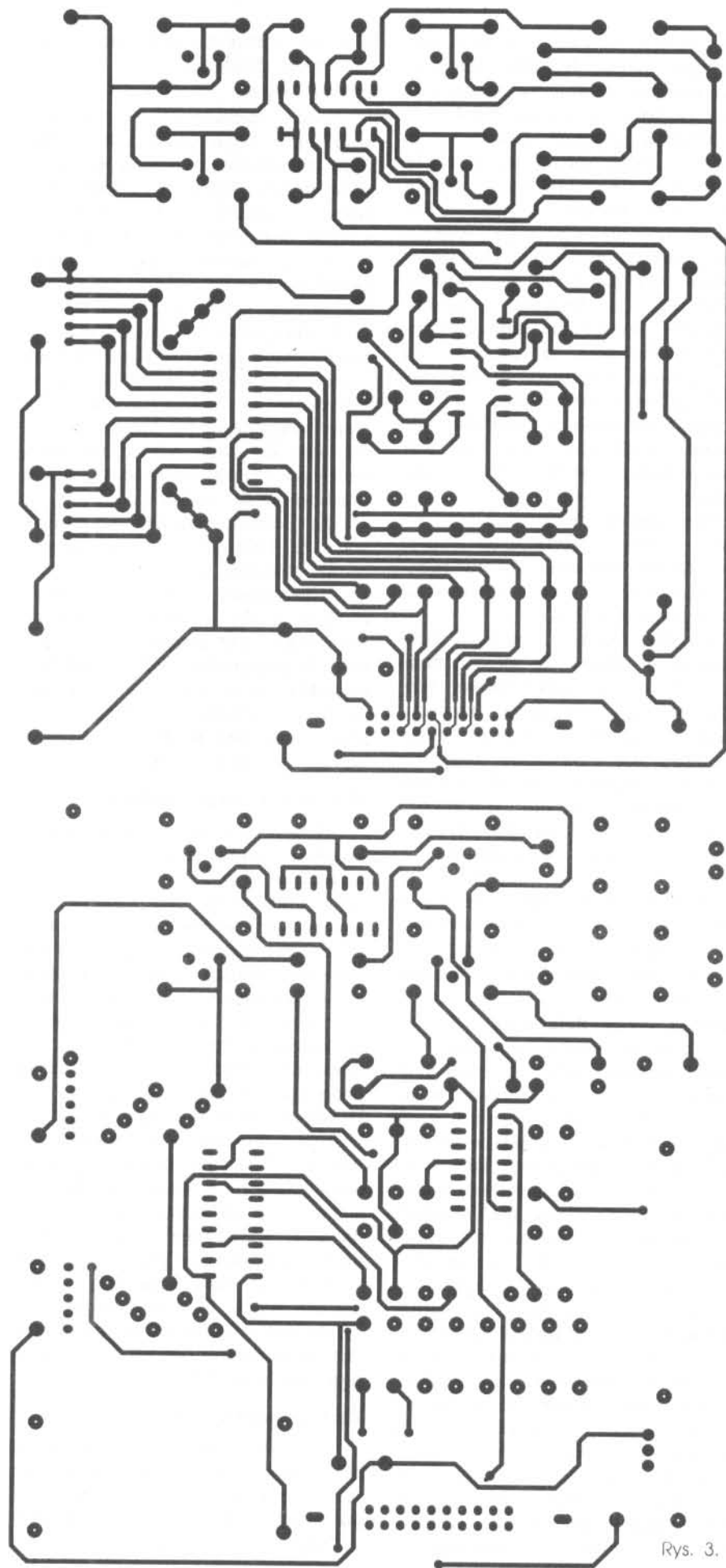
Dla ścisłości trzeba zasygnalizować, że komputerowe oprogramowanie sterujące mierzy raczej szybkość przesuwania niż liczy kroki (pomiar liczby kroków na sekundę - szybkość - jest bardziej ergonomiczny). W przypadku MAC'ia istnieje jednakże możliwość wyboru pomiędzy szybkością a ilością kroków, więc jest możliwe odwzorowanie rysunku w skali.

Montaż i uruchomienie

Montaż nie powinien sprawiać żadnych trudności, szczegóły można znaleźć na rys. 3 i 4. Należy raczej użyć podstawek pod układy scalone. Następnie trzeba przy pomocy omomierza odnaleźć wspólny punkt uzwojeń silnika. Przed montażem ULN2803 należy odszukać właściwe fazy do poprawnej detekcji przesunięcia. Pozostałe dwie można dowolnie połączyć, resztę wykona program. Trzeba pamiętać że ULN2803 musi otrzymać niski poziom aby tranzystory pozostawały odłączone. W trybie detekcji przesunięcia silnik krokowy działa jak generator i wytwarza nie pomijalną moc. Układ diod pozwala ograniczyć powstające przepięcia. Gdy napięcie to przekracza 12V, wewnętrzne diody ULN2803 zapewniają ograniczenie i nadmierny prąd zostaje skierowany do niestabilizowanego zasilacza 12V.

Oprogramowanie

Ponieważ sprzęg ten jest jedynie częścią urządzenia, nie można przedstawić kompletnego programu. Jak zwykle zostanie tylko zanalizowany sterownik (driver). Jest on pro-



Rys. 3.

gramem zapewniającym sterowanie uzwojeniami i poprawną współpracę z programem aplikacyjnym, który oczekuje użycia poleceń wysokiego stopnia, takich jak „ruch N kroków w kierunku X” albo „przejdź na detekcję przesunięcia”. Poniższe programy są przytoczone przykładowo, z biblioteki dla 68705P3.

Architektura oprogramowania jest identyczna jak do „FLY BY WIRE” (elektroniczny system sterowania nowoczesnymi samolotami). Sterowanie nie działa bezpośrednio, drążek sterowy nie jest połączony ze sterami samolotu, tylko jego położenie jest analizowane przez komputer. Określa on pozycje, w jakich powinny się znaleźć ster-y i wysyła przewodem (elektrycznym lub optycznym) polecenia do silników, mających zmienić pozycje sterów. Program działa zgodnie z tą zasadą. Polecenie jest analizowane po jego odebraniu i sygnalizowane są jego modyfikacje.

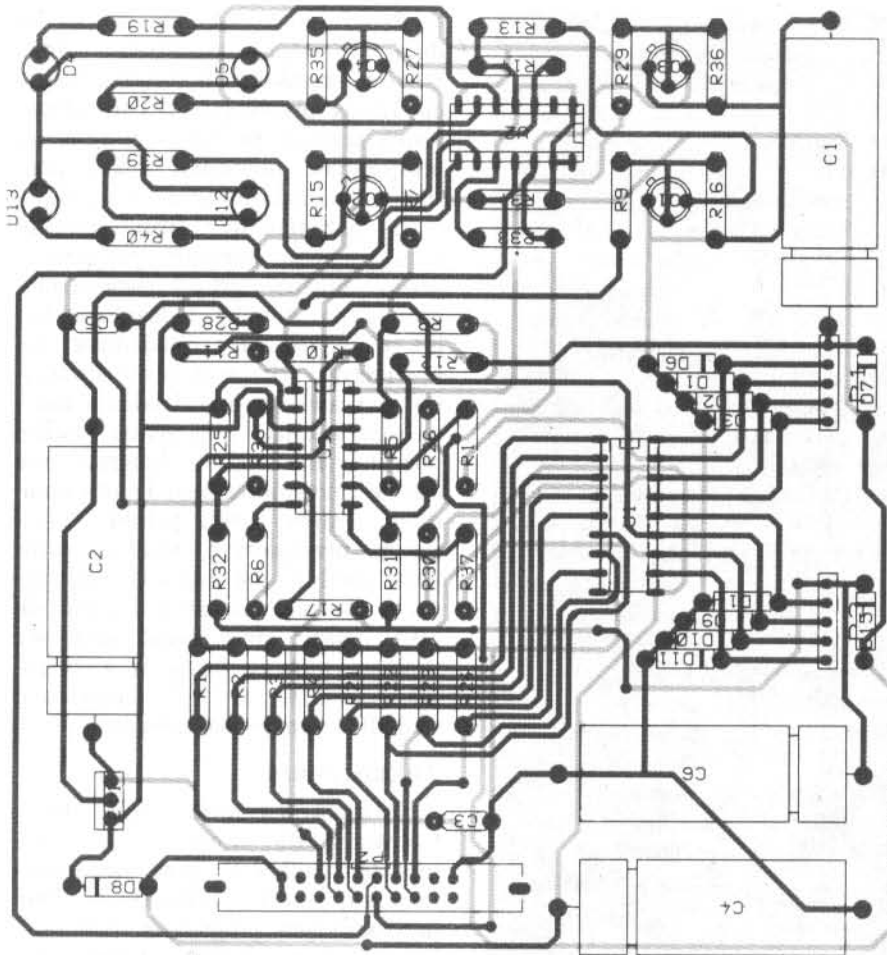
Detekcja przesunięcia

Nie będzie opisywana architektura programów z prywatnej biblioteki autora, z której jest wzięty ten moduł - zajęłoby to zbyt dużo miejsca. Pozwala on na szybką instalację sterownika. Aby go uruchomić, wystarczy aby etykieta FLECMOT miała wartość 1. IF FLECMOT rozpoczyna od rezerwacji miejsca na zmienne w RAM. Translacja warunkowa jest często używana, widać wiązanie etykiet w PPA.

```
IF FLECMOT          : @zmienna dla
                   : silnika przy
                   : detekcji
                   : przesunięcia
PPA: EQU FLAGG+1  : przesunięcie/
                   : rezerwuje się
                   : 1bajt
ELSE
PPA: EQU FLAGG    : nie rezerwuje
                   : się bajtu
ENDI
```

Program ten jest bardzo krótki, powinien być zarządzany przerwaniem TIMER, które jest zaimplementowane w prawie wszystkich rozwiązaniach (zarządza także interfejsem RS232).

W poniższym przykładzie PC0 i PC1 są przyłączone do wyjść komparatorów. Program aplikacyjny przeczyta D3 ze zmiennej PPA i D2, aby poznać kierunek przesunięcia i czy miało ono miejsce.



Rys. 4. Rozmieszczenie elementów na płytce drukowanej

```

IF FLECMOT ; *ODCZYT SILNIKA
; MT0 Z PC0, PC1
LECM0: LDA PORTC ; D1 D0
EOR PPA ; D0: WSKAŹNIK
PRED D0
RORA ; ZMIENIĆ?
BCS LECM01 ; TAK
LECM0F: BCLR 1,PPA
RTS
LECM01: BRCLR 0,PPA
LECM02
EOR #1
LECM02: RORA
BCC LECM03
BSET 3,PPA
SKIP2A ; POMIŃ A (omija
; się następną
; instrukcję)
LECM03: BCLR 3,PPA ; D3:
KIERUNEK PRZESUNIĘCIA
BSET 2,PPA ; D2: oznacza
; przesunięcie
BCLR 1,PPA ; D1: DO ZERA
INC PPA
BRA LECM0F
ENDI
    
```

A oto program testujący, wstawiony w pętlę programu głównego. Wysyła on do złącza szeregowego

literę „t” lub „p”, w zależności od kierunku ruchu (w tył albo w przód).

```

IF FLECMOT
TMOTEUR:BRSET2,PPA,TMOT1
; przy 1 sterownik
; wykrył przesunięcie
RTS
TMOT1: BCLR 2,PPA
; skasowanie flagi,
; ustalenie kierunku
; obrotu dla następnego
; przebiegu
LDA # „p”
BRSET 3,PPA,TMOT2
LDA # „t”
TMOT2: JMP OUTCAR
ELSE
TMOTEUR:RTS ; Jeżeli sterownik
; LECMOT nie został
; zdefiniowany
; i jeżeli program
; główny odwołuje
; się do niego,
; kieruje się nim...
ENDI
    
```

Trzeba zwrócić uwagę, że dla prostych aplikacji zarządzanie krytycznymi sekcjami nie jest analizowa-

ne, to też w tym przypadku można zapomnieć kroki, jeśli pętla programu głównego jest za długa; błąd o jedno p lub t za mało (lub za dużo) również może się pojawić.

Sterowanie silnikiem krokowym

Poniższy program pozwala sterować dwoma silnikami. Są one połączone z portem PORTA który powinien zostać skonfigurowany jako wyjście.

Kolejność faz jest zdefiniowana w tabeli. Jeżeli się zachowuje tę kolejność, należy odszukać właściwe fazy uzwojeń, w przeciwnym wypadku trzeba ją zmodyfikować.

Zadeklarowane są dwie tabele, jedna do napędzania silnika po pół kroku a druga po całym. W każdym bajcie tabeli jest:

- D0-D3: sterowanie FAZAMI
- D4: oznacza KIERUNEK SILNIKA
- D5-D6: TYP FAZY (krok, pół kroku)
- D7: NIE UŻYTY

Dla pół kroku tabela jest więc jak poniżej, trzeba jednak zwrócić uwagę, że występują w niej naprzemiennie jeden bit, a potem dwa bity o wartości 1, co charakteryzuje pół kroku.

```

;*09,01,03,02,06,04,12,08 kolejność faz.
TPHASE: DFB 01,03,06,02 ;
DFB 12,01,04,01 ; Wartość fazy
DFB 09,01,01,01 ; bieżącej jest
DFB 08,01,01,01 ; wzięta jako
DFB 11H,19H,13H,11H ; adres tabeli,
DFB 16H,11H,12H,11H ; jej zawartość
DFB 1CH,18CH,11H,11H ; daje nową
DFB 14H,11H,11H,11H ; fazę
    
```

Do odwrócenia kierunku wystarczy przejść do tej drugiej części tabeli.

W przypadku całego kroku:

```

;*09,03,06,12
DFB 23H,23H,23H,26H ; W przypadku
DFB 23H,23H,2CH,23H ; całego kroku,
DFB 23H,23H,23H,23H ; aby mieć
DFB 29H,23H,23H,23H ; tabele tej
DFB 33H,33H,33H,39H ; samej wiel-
DFB 33H,33H,33H,33H ; kości i dla
DFB 33H,3CH,33H,33H ; łagodnego
DFB 36H,33H,33H,33H ; przejścia od
; trybu „pół
; kroku” do
; trybu „cały
; krok”,
; dodano fazy
; uzupełniające
    
```

Program sterowania silnikami zapewnia sterowanie szybkością i pozycjonowaniem.


```

;@VARIABLES
MOPHASE: EQU COMMDE+1
        ; do sterowania silnikami
M0STATUS: EQU M0PHASE+1
MOVIT: EQU M0STATUS+1
MOVITC: EQU M0VIT+1
PBPHASE: EQU M0VITC+1
MONBPL: EQU PBPHASE+1
MONBPH: EQU M0NBPL+1
    
```

Tablica ta stosuje się do jednego silnika. Dla drugiego silnika struktura odpowiedniej tablicy jest taka sama. Rejestr X zapewnia poprawne wskazywanie parametrów RAM silnika 0 lub 1; przy każdym przywołaniu uaktualnia on dane w PORTA.

Zmienna PBPHASE zapamiętuje uaktualnienie pierwszego silnika, podczas gdy program GMOTEUR wykonuje obliczenie nowej fazy dla drugiego.

```

GSMOTE: LDX #0 ; pierwszy silnik
JSR GMOTEUR
LDA M0PHASE,X
AND #0FH
STA PBPHASE ; MAJ PB
LDX #8 ; drugi silnik
JSR GMOTEUR ; D4-D7
LDA MOPHASE,X
LSLA
LSLA
LSLA
LSLA
ORA PBPHASE ; M1,M2
STA PORTA ; uaktualnienie
        ; portów
RTS
    
```

Poniższy fragment programu jest krótki, może być przywoływany przez IT albo wstawiony do programu głównego. W pierwszym przypadku praca silnika będzie niewątpliwie bardziej regularna, trzeba jednak uważać, aby nie przeciążyć zarządzania IT. Rejestr X wskazuje deskryptor.

Ten program steruje szybkością, pozycją oraz detektorem końca przebiegu.

```

GMOTEUR: LDA M0STATUS,X
        ; rozpoczyna się sprawdzeniem
        ; czy silnik jest aktywny
BMI GMOTEUR0 ; zerowy 8 BIT
M0STATUS ; oznacza postój
RTS ; ...
GMOTEUR0: LDA M0VITC,X
        ; Ten rejestr definiuje
        ; szybkość silnika i czeka
        ; o ile nie jest zerowa.
BEQ GMOTEUR1
DEC M0VITC,X
RTS
GMOTEUR1: LDA M0PHASE,X
        ; uaktualnienie faz,
    
```

```

STX TT ; odczyt tabeli, TT jest
        ; komórką pamięci
        ; „tymczasowej”.
TAX
LDA TPHASE,X
LDX TT
STA M0PHASE,X
LDA M0STATUS,X
        ; Czy trzeba
        ; uwzględnić
        ; detektor końca
        ; przebiegu?
AND #20H ;
BEQ GMOTEU11 ; = 0 nie jeśli zero
LDA M0STATUS,X ; koniec przebiegu?
AND #40H
BNE GMOTEUR ; tak, koniec także
        ; na MXNBP = 0!
GMOTEU11: LDA M0NBPL,X
        ; sterowanie ilością
        ; kroków
BNE GMOTEUR3
LDA M0NBPH,X ; koniec?
BNE GMOTEUR2
GMOTEURF: LDA M0STATUS,X
        ; Zatrzymanie silnika
AND #7FH ; Wyzerowanie D7
STA M0STATUS,X ; OK
AND #10H ; POWER off na
        ; końcu przebiegu?
BEQ GMOTEUF ; tak
RTS
GMOTEUF: LDA M0PHASE,X
        ; wyłączenie prądu
        ; w uzwojeniach
AND #0F0H
STA M0PHASE,X ; ale gubi się
        ; bieżącą fazę
RTS
GMOTEUR2: DEC M0NBPH,X
GMOTEUR3: DEC M0NBPL,X
LDA MOVIT,X ; uaktualnienie
        ; licznika szybkości,
        ; im MOVIT STA
        ; MOVIT, jest większe
        ; tym mniejsza jest
        ; szybkość
RTS
    
```

Przed uruchomieniem silnika łąduje się różne parametry: ilość kroków, szybkość, polecenie detekcji końca przebiegu. W celu sterowania końcem przebiegu zewnętrzny podprogram powinien uaktualnić stan. Po załadowaniu wszystkich tych parametrów, uruchomienie silnika może nastąpić po prostu przez zmianę wartości ósmego bitu statusu na jedynkę. Odczytując ten bit można również dowiedzieć się, czy program zatrzymał silnik. Przyczyną nieprawidłowego działania można ustalić odczytując status i liczbę pozostałych kroków. Program nie steruje przyspieszaniem i spowalnianiem silnika, jeżeli jednak chce się go przyspieszyć, istnieje możliwość modyfikacji szybkości w trakcie działania. Przed zatrzymaniem silnika należy zredukować szybkość; program główny może ją zredukować z chwilą, gdy wartość MONBPH stanie się zerem.

Zakończenie

Program ten może zostać przystosowany do innego chipu, na przykład w Basicu, przy zastosowaniu wyjścia dla drukarki i wyżej przytoczonych programów (ale ilość jego wierszy będzie z pewnością większa).

Umożliwia on działanie w trybach „cały krok“, „pół kroku“, ze zmienną szybkością i zatrzymywaniem się po zadanej ilości kroków albo po wykryciu końca przebiegu. Po zatrzymaniu się może wyłączyć prąd albo utrzymywać silnik pod napięciem. W razie potrzeby, gdy nie są potrzebne wszystkie funkcje, może zostać uproszczony.

ERP