

Dwukierunkowy regulator obrotów silników DC

AVT-5011



Regulatory obrotów silników prądu stałego są nieśmiertelnym tematem w pismach przeznaczonych dla elektroników, przede wszystkim ze względu na wielość możliwych aplikacji takich urządzeń.

Tym razem proponujemy mikroprocesorowy regulator obrotów, w którym klasyczny potencjometr zastąpiono nastawnikiem impulsowym.

Bardzo interesujące rozwiązanie!

W zasadzie sprawa jest prosta: wystarczy zbudować generator impulsów prostokątnych o stałej częstotliwości i zmiennym wypełnieniu (np. na NE555) i dołączyć do jego wyjścia tranzystor wykonawczy. Sprawa zaczyna się jednak komplikować, jeżeli regulować mamy nie tylko prąd pobierany przez silnik, ale i kierunek jego obrotów. Jeżeli do tego założymy, że zarówno prędkość obrotowa, jak i kierunek obrotów mają być zmieniane za pomocą jednego pokrętkła, z zatrzymaniem silnika w jego neutralnym położeniu, to całość zaczyna się komplikować. Tego nie da się już zrealizować na jednym typowym układzie scalonym, szczególnie jeżeli warunkiem jest stosowanie wyłącznie półprzewodnikowych elementów przełączających, bez jakichkolwiek przekaźników.

Regulator obrotów silników prądu stałego znajdzie z pewnością zastosowanie przy zasilaniu silników małych obrabiarek czy

wiertarek, stosowanych np. w modelarstwie. Z praktyki wiem, że sterowanie takimi urządzeniami wyłącznie za pomocą potencjometru regulującego obroty silnika i zatrzymującego jego ruch w krańcowym położeniu jest dość kłopotliwe. Za każdym bowiem razem, po ponownym uruchomieniu obrabiarki musimy „odnajdywać” właściwą, uprzednio występującą prędkość obrotową napędzającego silnika. A zatem dobrze byłoby zapamiętywać ostatnio ustaloną prędkość obrotową i powracać do niej po powtórny włączeniu silnika. Dobrze by też było, aby projektowany układ mógł pracować w dwóch trybach: jednym z dokładną regulacją, powiedzmy z krokiem 1%, i w drugim zgrubną, np. z krokiem co 10%.

Te wszystkie uwarunkowania spowodowały, że do sterowania pracą regulatora postanowiłem zastosować procesor, ale procesor dość nietypowy. Po raz kolejny postanowiłem skorzystać

z pomocy mojego faworyta: „maleńkiego“ AT TINY, mikroprocesorka o ośmiu wyprowadzeniach (pięciu uniwersalnych), obywatelkiego się nawet bez zewnętrznego rezonatora kwarcowego. Niech jednak nikogo nie zmyli przydomek „tiny“ nadany przez producenta temu układowi: jest to pełnowartościowy procesor nie ustępujący w niczym poza liczbą wyprowadzeń i nieco zubożonym wewnętrznym hardwarem, swoim braciom z rodziny AVR.

Proponowany układ, głównie dzięki zastosowaniu procesora, jest banalnie prosty i łatwy do wykonania nawet dla zupełnie początkującego elektronika. Jego zalety praktyczne zostały przeze mnie potwierdzone podczas długotrwałego używania regulatora do sterowania obrotami miniaturowej wiertarko-frezarki, stosowanej podczas budowy plastikowych modeli samolotów.

Opis działania

Schemat elektryczny mikroprocesorowego regulatora obrotów silników DC pokazano na rys. 1. Sercem układu jest tu zaprogramowany procesor typu AT TINY lub AT90S2313, do którego aktywnych wyprowadzeń dołączono proste układy wykonawcze i sterujące. Jako elementy wykonaw-

cze, bezpośrednio zasilające dołączony do regulatora silnik, zastosowałem cztery tranzystory MOSFET, pracujące w układzie mostka, w którego przekątnej został włączony silnik. Należy zauważyć, że przewodzenie sterowanego przez procesor tranzystora T3 spowoduje jednoczesne włączenie tranzystorów T1 i T6 i obrót silnika w kierunku wskazówek zegara (oczywiście umownie!). Podobnie, wysterowanie bazy tranzystora T4 wywoła włączenie tranzystorów T2 i T5 i obrót silnika w kierunku przeciwnym do ruchu wskazówek zegara. Lepiej nawet nie myśleć, jakie skutki wywoła jednoczesne włączenie tranzystorów T3 i T4, ale ci Koledzy, którzy zechcą samodzielnie napisać program sterujący regulatorem, powinni zadbać, aby taka sytuacja nigdy się nie wydarzyła. W przeciwnym wypadku, jak mówił Grek Zorba, spotka nas „piękna katastrofa“.

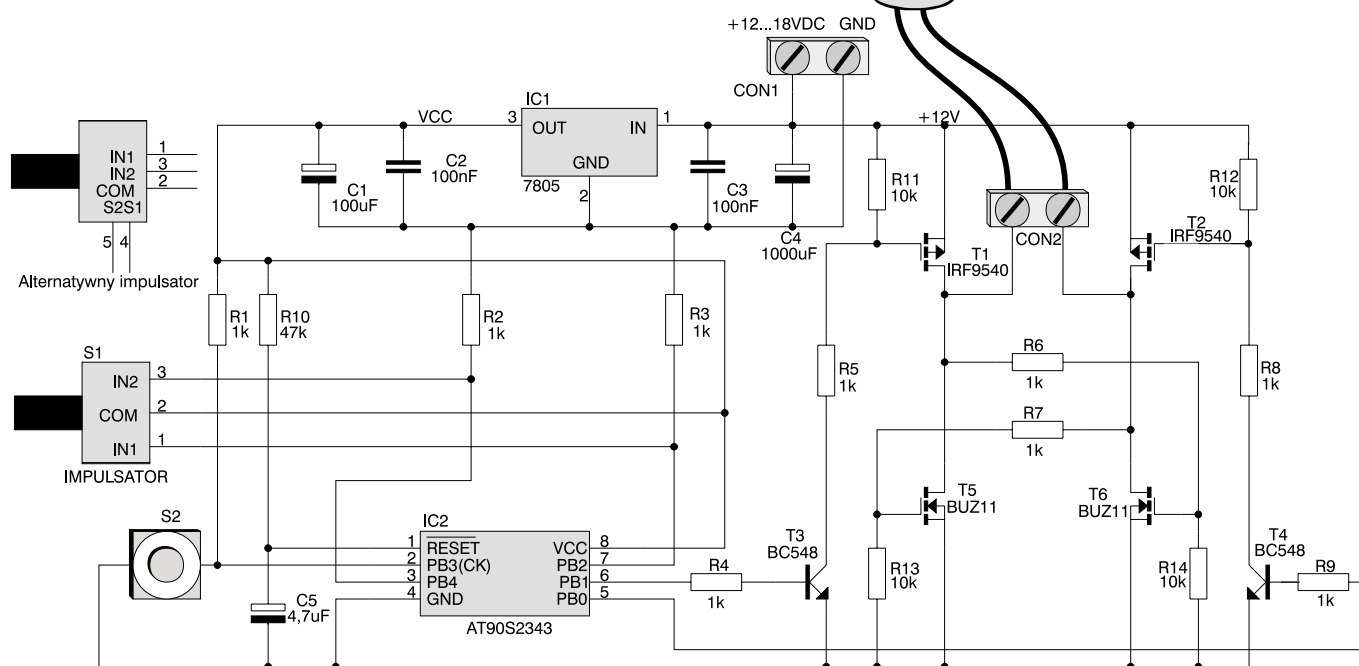
Czytelnicy, którzy obejrzeni już zdjęcia przedstawiające model układu regulatora z pewnością są przekonani, że nasz regulator będzie sterowany za pomocą potencjometru. Element umieszczony po prawej stronie płytki rzeczywiście do złudzenia przypomina potencjometr, ale tak naprawdę nie ma z tym powszechnie znanym elementem nic wspólnego.

Tab. 1. Stany na wyjściach impulsatora w zależności od kierunku obracania ośki.

	Obrót w prawo		Obrót w lewo	
	Styk 1	Styk 2	Styk 1	Styk 2
Krok1	1	0	0	1
Krok2	1	1	1	1
Krok3	0	1	1	0
Stop	0	0	0	0

go. S1 jest obrotowym impulsatorem mechanicznym (znanym już z projektów publikowanych w Elektronice Praktycznej) produkowanym przez firmę Bourns, który podczas obracania jego ośką generuje impulsy na dwóch wyjściach. Sekwencja tych impulsów (tab. 1) jest tak dobrana, że dołączony do wyjść impulsatora procesor może z łatwością nie tylko liczyć impulsy, ale także określić kierunek obrotu ośki impulsatora.

Zastosowanie impulsatora obrotowego zamiast zwykle używanych przycisków znacznie zwiększyło komfort obsługi regulatora.



Rys. 1. Schemat elektryczny regulatora obrotów.

Za pomocą impulsatora nie tylko możemy wygenerować dowolną liczbę impulsów, ale łatwo, intuicyjnie zwiększać lub zmniejszać ich częstotliwość, przechodząc ze zgrubej regulacji na precyzyjną.

Pozostała część układu to już tylko typowy zasilacz, dostarczający napięcia +5VDC, niezbędnego do zasilania procesora. Ze względu na chęć maksymalnego uproszczenia regulatora i zmniejszenia kosztów jego wykonania nie zastosowałem w nim typowego układu zerującego procesor DS1813, a zastąpiłem go prostym obwodem z kondensatorem C5 i rezystorem R10. To chyba wszystko, co można napisać o tak prostym hardware. Zajmijmy się teraz sterującym nim programem.

Po wykonaniu typowych czynności konfiguracyjnych, z których najważniejszą jest określenie sposobu pracy Timera0:

```
Config Timer0 = Timer, Prescale
= 8
Start Timer0
Enable Timer0
On Timer0 Tim_int
```

oraz zbadanie stanu styków przycisku S2 podczas włączenia zasilania:

```
Reset Portb.3
If Pinb.3 = 1 Then
  Step_flag = 1
Else
  Step_flag= 0
End If
```

program wchodzi w pętlę podprogramu REGULATION, w której już będzie pracował aż do wyłączenia zasilania.

Pierwszą czynnością jaka zostanie wykonana w pętli jest sprawdzanie stanu styków impulsatora oraz przycisku S2:

```
Sub Regulation
Do
Reset Portb.4: Reset Portb.2
'spróbuj ustawić stany niskie
'na wyjściach impulsatora
If Pinb.4 = 1 And
  Pinb.2 = 0 Then
'jeżeli na wyjściu dołączonym
'do pinu 4 portu B pojawił się
'stan wysoki, czyli impulsator
'rozpoczął obrót w lewo, to:
```

```
Do
If Pinb.2 = 1 Then Exit Do
'zaczekaj w pętli aż do
'wykonania przez impulsator
'dalszej części obrotu
Loop

Do
If Pinb.4 = 0 Then Exit Do
'następnie zaczekaj na
'wykonanie przez impulsator
'pełnego kroku
Loop

Incr Regulation_counter

If Step_flag = 1 Then
  Regulation_counter =
    Regulation_counter + 9
'jeżeli układ pracuje w trybie
'regulacji zgrubej to dodatkowo
'zwiększ wartość tej zmiennej
'o 9
If Regulation_counter > 100
  Then Regulation_counter = 100
End If
```

Podczas pracy programu w pętli badana jest wartość zmiennej REGULATION_COUNTER. Należy tu zaznaczyć, że zmienna ta została zadeklarowana jako INTEGER, czyli że może przyjmować wartości ujemne, w zakresie od -32768 do +32767. Jeżeli program wykryje, że zmienna ta przyjęła wartość 0, to:

```
If Regulation_counter = 0 Then
  Reset Portb.0: Reset Portb.1
'wyłącz tranzystory sterujące
'stopniem mocy
End If
```

A więc, podczas pokręcania ośką impulsatora i zmniejszania obrotów silnika, obojętnie w jakim kierunku pracowałyby, trafiamy w pewnym momencie na „położenie neutralne“, w którym silnik zostaje wyłączony.

Zmienna REGULATION_COUNTER badana jest także pod kątem jej wartości względem zera:

```
If Regulation_counter > 0 Then
  Direction_flag = 1
'wskaźnik kierunku obrotu
'ustawiany jest na 1
Else
  Direction_flag = 0
'wskaźnik kierunku obrotu
'ustawiany jest na 0
End If
```

```
If Regulation_counter <> 0 Then
  Regulation_flag = 1
'wskaźnik włączenia silnika
'ustawiany jest na 1
  Enable Interrupts
Else
  Regulation_flag = 0
'wskaźnik włączenia silnika
'ustawiany jest na zero
  Disable Interrupts
End If
```

Ważną rolę w sterowaniu pracą układu odgrywa przycisk S2, którego naciśnięcie może powodować dwie odmienne reakcje programu:

```
Reset Portb.3
'spróbuj ustawić stan niski
'na pinie 3 portu B
If Pinb.3 = 1 And
  Regulation_flag = 1 Then
'jeżeli próba nieudana
'i wskaźnik włączenia silnika
'ustawiony jest na 1, to:
  Disable Interrupts
  Writeeprom Regulation_counter,1
'zapisz w pamięci EEPROM
'aktualną wartość zmiennej
'REGULATION_COUNTER
  Reset Portb.0: Reset Portb.1
'wyłącz silnik
  Regulation_counter = 0
  Regulation_flag = 0
'ustaw wskaźnik włączenia
'silnika na zero
  Wait 1
  Enable Interrupts
End If
```

'Dodatkowego komentarza wymaga 'powód zawieszania obsługi 'przerwania podczas wykonywania 'procedury zapisu lub odczytu 'danych z pamięci wewnętrznej 'EEPROM procesora. Niewyłączenie obsługi przerwania 'pochodzącego od timera może 'powodować błędy w operacjach 'dokonywanych na pamięci EEPROM

```
Reset Portb.3
'spróbuj ustawić stan niski
'na pinie 3 portu B
If Pinb.3 = 1 And
  Regulation_flag = 0 Then
'jeżeli próba nieudana
'i wskaźnik włączenia silnika
'ustawiony jest na 0, to:
  Disable Interrupts
  Readeeprom Temp2, 1
'odczytaj z pamięci EEPROM
'poprzednią wartość zmiennej
```

```
'REGULATION_COUNTER
  If Temp2 > 0 Then
    Direction_flag = 1
'wskaźnik kierunku obrotów jest
'ustawiany na 1
    Regulation_counter = 1
  End If

  If Temp2 < 0 Then
    Direction_flag = 0
'wskaźnik kierunku obrotów jest
'ustawiany na 0
    Regulation_counter = -1
  End If

  Soft_start_flag = 1
'znacznik zezwolenia na
'realizację miękkiego startu
'jest ustawiany na 1

Regulation_flag = 1
'znacznik włączenia regulacji
'jest ustawiany na 1

Waitms 255 'zaczekaj 255 ms
Enable Interrupts

End If
```

Zastosowany w układzie procesor jest nieco uproszczony w stosunku do innych, „większych“ procesorów z rodziny AVR. Wyposażony jest tylko w jeden timer, który nie dysponuje możliwością sprzętowej realizacji PWM. A zatem musimy jej dokonać programowo, a w pętli głównej programu dokonywane są wstępne obliczenia wartości zmiennych decydujących o współczynniku PWM.

```
Regulation_value =
  Abs(regulation_counter)
'przeliczenie zmiennej
'REGULATION_COUNTER na jej
'wartość bezwzględną
Regulation_value =
  Regulation_value * 2.5
'dostosowanie obliczonej
'wartości do pojemności
'rejestru timera0

Pw1 = Regulation_value
'obliczenie pierwszej
'wartości PWM

Pw2 = 255 - Pw1
'obliczenie drugiej
'wartości PWM
```

Raptowne włączanie zasilania nie jest „zdrowe“ ani dla silni-

ków, ani dla zasilających je źródła energii. Dlatego też w programie została zaimplementowana procedura miękkiego startu silnika, realizowana po naciśnięciu przycisku S2 i odczytaniu z pamięci EEPROM zapisanego tam współczynnika regulacji.

```
If Soft_start_flag = 1 Then
'jeżeli wskaźnik aktywności
'miękkiego startu został
'ustawiony na 1, to:
If Direction_flag = 1 Then
  Incr Regulation_counter
'jeżeli wskaźnik kierunku
'obrotów ustawiony jest na 1,
'to zwiększ o 1 wartość
'zmiennej REGULATION_COUNTER

If Direction_flag = 0 Then
  Decr Regulation_counter

If Regulation_counter = Temp2 Then
  Soft_start_flag = 0
'jeżeli zmienna REGULATION_
'COUNTER osiągnęła wartość
'odczytaną z pamięci EEPROM,
'to koniec procedury
'miękkiego startu

Waitms 10 'zaczekaj 10 ms
End If
```

Jak dotąd, zadaniem wszystkich działań podejmowanych przez program sterujący pracą układu było jedynie przygotowanie danych, które zostaną wykorzystane do regulacji mocy silnika metodą PWM. To, co najważniejsze dla działania układu, rozgrywa się w małym podprogramie TIM0_INT, wykonywanym po zgłoszeniu przerwania od Timera0.

```
Tim0_int:

Stop Timer0
'zatrzymaj pracę Timera0
X = Not X
'zmienna pomocnicza X,
'deklarowana jako bit
przyjmuje
'przeciwną wartość
If X = 1 Then
  Timer0 = Pw2
'załaduj do rejestru Timera0
'obliczoną wartość PW2

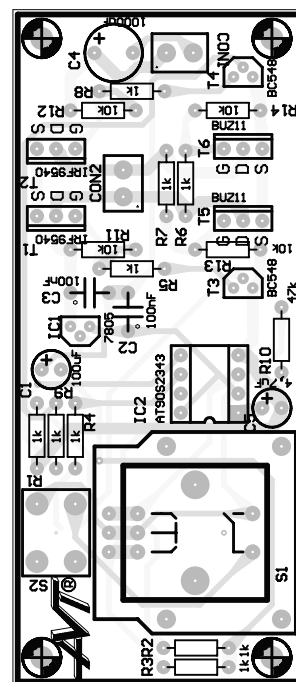
If Direction_flag = 1 Then
'jeżeli silnik ma obracać się
'w prawo (umownie), to:
```

```
Set Portb.0
'włącz tranzystor T4
Else
  Set Portb.1
'włącz tranzystor T3
End If

Else
  Timer0 = Pw1
  Reset Portb.0: Reset Portb.1
'wyłącz obydwa tranzystory
'sterujące
End If
Start Timer0
'uruchom TIMER0
Return
```

Montaż i uruchomienie

Na rys. 2 pokazano rozmieszczenie elementów na płycie obwodu drukowanego, wykonanego na laminacie dwustronnym z metalizacją. Montaż tak prostego układu w zasadzie nie wymaga komentarza, ale chciałbym poruszyć dwie istotne sprawy. Pierwsza dotyczy sposobu zamocowania impulsatora i przycisku S2. Elementy te mogą zostać przylutowane do płytki z dowolnej strony, z tym, że zalecałbym raczej zamontowanie ich od (umownej) strony ścieżek. Taki sposób montażu ułatwi umieszczenie układu w obudowie, z której wyprowadzona zostanie ośka impulsatora i końcówka przycisku S2.



Rys. 2. Rozmieszczenie elementów na płycie drukowanej.

WYKAZ ELEMENTÓW**Rezystory**

R1..R9: 1kΩ

R10: 47kΩ

R11..R14: 10kΩ

Kondensatory

C1: 100μF/10V

C2, C3: 100nF

C4: 1000μF/25V

C5: 4,7μF/16V

Półprzewodniki

IC1: 7805

IC2: AT90S2343

T1, T2: IRF9540

T3, T4: BC548

T5, T6: BUZ11

Różne

CON1, CON2: ARK2 (3,5mm)

S1: impulsator obrotowy

S2: przycisk microswitch
(opcjonalnie)

Drugi problem dotyczy samego impulsatora. W prototypie pokazanym na fotografii zastosowany

został impulsator produkcji firmy Bourns o dość sporych wymiarach. Otrzymałem ostatnio informacje, że mogą wystąpić problemy z zaopatrzeniem się w przyszłości w taki właśnie typ impulsatora. Dlatego też płytką obwodu drukowanego została przystosowana także do montażu impulsatora innego typu (firmy Piher), który zawiera już w sobie także przycisk potrzebny do sterowania pracą układu. Stosując drugi typ impulsatora, nie musimy montować przycisku S2. Jego rolę przejmie ośka impulsatora, która może być nie tylko obracana, ale także naciskana. Zwiera wówczas wbudowane w impulsator dodatkowe styki.

Układ zmontowany ze sprawdzonych elementów nie wymaga żadnej regulacji i po włożeniu zaprogramowanego procesora w podstawkę działa natychmiast poprawnie. Jeżeli w momencie włączenia zasilania przytrzymamy przez chwilę przycisk S2, to

układ rozpocznie pracę w trybie regulacji zgrubej, tj. ze skokiem co 10%.

Podczas pracy regulatora każde naciśnięcie przycisku S2 powoduje natychmiastowe zatrzymanie silnika i zapamiętanie zarówno jego mocy, jak i kierunku obrotów. Silnik można powtórnie uruchomić dwoma sposobami:

1. Ponowne naciśnięcie przycisku S2 spowoduje włączenie silnika z zapamiętanymi parametrami jego pracy.

2. Obracanie ośką impulsatora spowoduje rozpoczęcie pracy silnika z parametrami zależnymi od kierunku jej obrotu i liczby wygenerowanych przez impulsator impulsów.

Zbigniew Raabe, AVT
zbigniew.raabe@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/maj01.htm> oraz na płycie CD-EP05/2001B w katalogu PCB.