

Stroboskopowy miernik prędkości obrotowej



Prezentowane urządzenie jest miernikiem prędkości obrotowej, a jego głównymi zaletami są możliwość wykonania pomiaru bez demontażu komponentów, nieskomplikowana obsługa i budowa. Zasada pomiaru sprowadza się do oświetlenia wirującego obiektu za pomocą błysków diody LED dużej mocy i takiej regulacji ich częstotliwości, aby obiekt wirujący wydawał się nieruchomy. Wówczas na wyświetlaczu jest wskazywana wartość obliczona na podstawie ustawionej pulsacji (częstotliwości błysków), która jest równoważna mierzonej prędkości obrotowej.

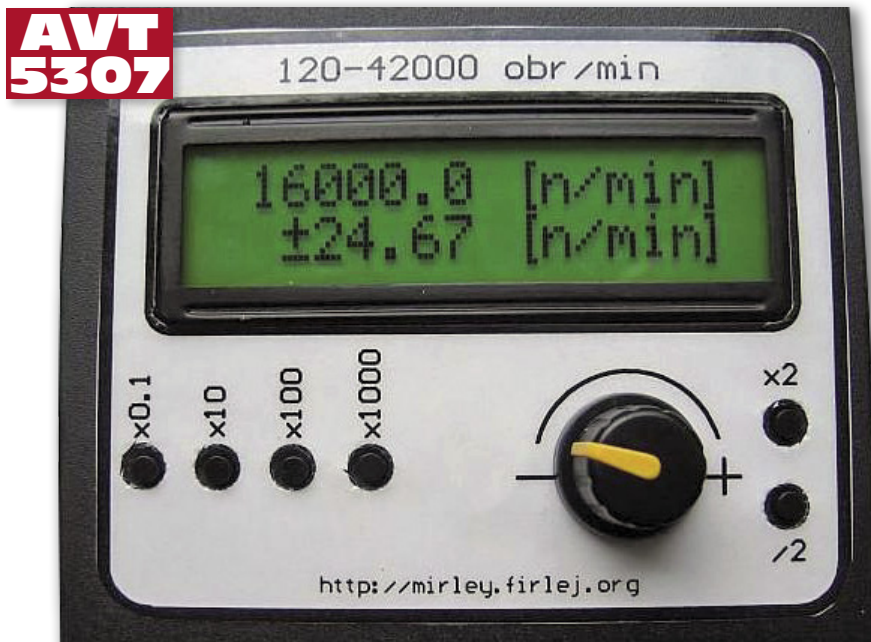
Rekomendacje: miernik przyda się w warsztacie mechanicznym, samochodowym oraz przy testowaniu i uruchamianiu układów sterujących silnikami elektrycznymi.

Opis działania układu

Schemat ideowy miernika obrotów pokazano na **rysunku 1**. Sercem układu i jego najważniejszą częścią jest mikrokontroler U1 (ATmega8-16AU) taktowany częstotliwością 16 MHz. Jest on programowany poprzez złącze **Prog** służące do programowania szeregowego. Jest ono konieczne, gdyż mikrokontroler ma obudowę SMD, co uniemożliwia jego łatwe zaprogramowanie w układzie zewnętrznym.

Zadaniem mikrokontrolera jest wygenerowanie przebiegu prostokątnego (o małym wypełnieniu), który zostanie potem przetworzony na błyski świetlne. Częstotliwość błysków musi być regulowana w szerokim zakresie, aby zakres pomiarowy też był duży. Ustawiona częstotliwość błysków jest przeliczana i prezentowana na wyświetlaczu LCD w postaci obrotów na minutę, co zwiększa komfort użytkownika układu.

Kondensator C5 (100 nF) filtruje zasilanie mikrokontrolera. Kondensatory C6 i C7 (100 nF) łagodzą zbrocza sygnału wygene-



rowanego przez impulsator IMP, co ułatwia jego bezproblemową obsługę w programie. Przyciski S1...S6 stanowią dodatkową klawiaturę urządzenia. Impulsator wraz z dodatkowymi przyciskami pozwala na łatwe ustawienie żądanej częstotliwości błysków. Ruch impulsatora zmienia mierzone obroty o ± 1 , gdy żaden przycisk mnożnika nie jest wciśnięty. Przytrzymanie jednego z przycisków ($\times 0.1$, $\times 10$, $\times 100$, $\times 1000$) podczas obracania impulsatorem powoduje, że obroty są zmieniane o ± 0.1 , ± 10 itd.

Elementem generującym błyski świetlne jest dioda LED o mocy 0,5 W. Jej prąd jest ograniczany przez rezystor R4 o mocy 2 W. Dioda jest sterowana za pomocą tranzystora T2. Jest ona włączona bezpośrednio pomiędzy bieguny zasilania, z pominięciem stabilizatora, aby zminimalizować wpływ impulsów prądowych na działanie mikrokontrolera i odciążyć stabilizator napięcia U2.

Prezentacja wyników pomiaru odbywa się na wyświetlaczu W1 (LCD 16 \times 2). Do regulowania jest kontrastu służy potencjometr P1 (10 k), a podświetlenie może być załączane przez tranzystor T1 (BC556). Ta wersja programu nie obsługuje jednak takiej funkcjonalności i podświetlenie jest przez cały czas załączone.

Budowa i uruchomienie

Na **rysunku 2** przedstawiono schemat montażowy miernika obrotów. Układ jest

AVT-5307 w ofercie AVT:
AVT-5307A – płytka drukowana
AVT-5307B – płytka drukowana + elementy

- Podstawowe informacje:**
- Pomiar prędkości obrotowej w zakresie 120...42000 obrotów/min,
 - Mikrokontroler ATmega8 taktowany sygnałem o częstotliwości 16 MHz,
 - Ustawione obroty są prezentowane na wyświetlaczu LCD,
 - Prezentacja błędów wynikających z zaokrąglenia wyliczonych w programie wartości dla podzielnika timera,
 - Sterowanie za pomocą enkodera obrotowego i sześciu przycisków,
 - Płytkę dopasowano do popularnej obudowy KM35,
 - Zasilanie z baterii 9 V (6F22).

Dodatkowe informacje:
http://mirley.firlej.org/stroboskopowy_miernik_obr

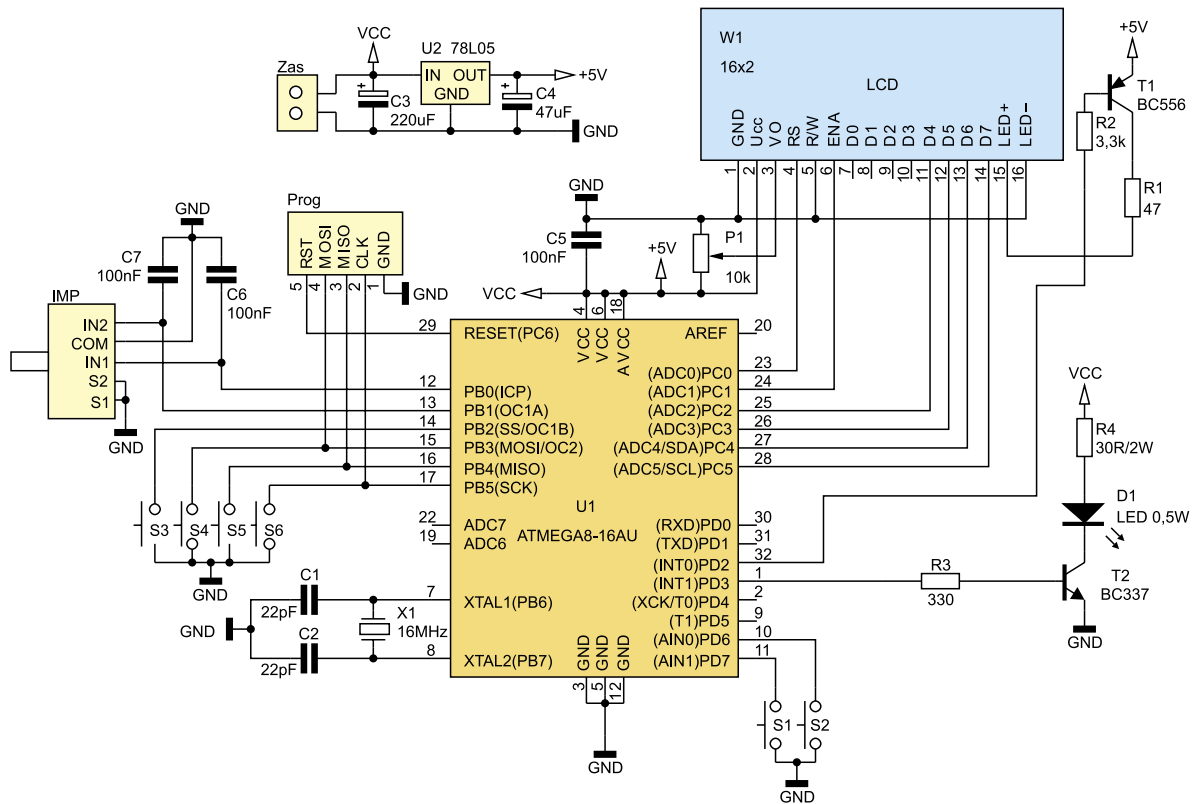
Dodatkowe materiały na CD/FTP:
<ftp://ep.com.pl>, user: 12040, pass: 15735862

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w **Wykazie elementów** kolorem czerwonym

Projekty pokrewne na CD/FTP:
(wymienione artykuły są w całości dostępne na CD)

- AVT-5260 Obrotomierz cyfrowy (EP 10/2010)
- AVT-2799 Mikroprocesorowy obrotomierz stroboskopowy (EdW 9/2006)
- AVT-434 Komputer samochodowy (EP 9-10/2005)
- AVT-2711 Obrotomierz (EdW 2/2004)

nieskomplikowany, ale jest zbudowany głównie z elementów SMD, których przyłutowanie może sprawić problemy początkującym. Montaż należy rozpocząć od wlutowania



Rysunek 1. Schemat ideowy obrotomierza

wania dwóch zworek. Następnie w kolejności powinny być kondensatory i rezystory SMD. Kolejnym krokiem jest przyłutowanie mikrokontrolera U1, przy którym należy zwrócić uwagę na numery wyprowadzeń. Jakakolwiek pomyłka przy lutowaniu mikrokontrolera może doprowadzić do jego uszkodzenia. Źle przyłutowany mikrokontroler może być bardzo trudny do odlutowania bez specjalistycznego sprzętu.

Osie przycisków powinny mieć długość ok. 15 mm. Należy je przyłutować w taki sposób, aby nieco wystawały ponad powierzch-

nię wyświetlacza LCD. Będzie to ważne przy montowaniu płytki w obudowie. Podobnie sprawa wygląda z impulsatorem. Potencjometr P1 musi być w obudowie stojącej, aby z boku można było regulować kontrast przy włożonym LCD.

Płytkę zaprojektowano tak, aby pasowała do popularnej obudowy KM35. Przy wycinaniu otworów w obudowie można posłużyć się **rysunkiem 3** (dostępny na płycie obudowy za pomocą taśmy klejącej i wiercić przez rysunek. Uwaga: na rysunku znajduje się podziałka. Jeśli wynik jej pomiaru będzie różny od podanego, należy odpowiednio przeskalować rysunek podczas wydruku.

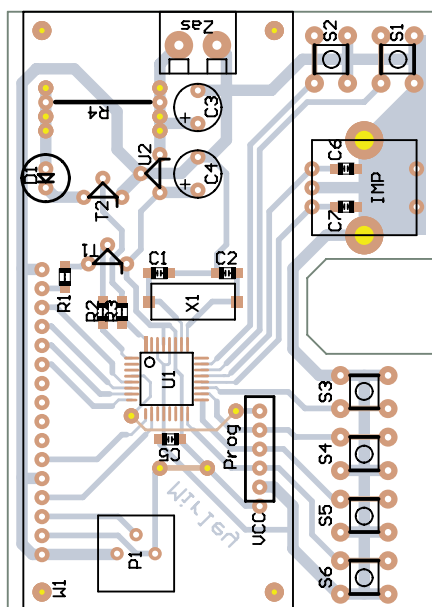
Układ modelowy jest zasilany z baterii 9 V, jednak nic nie stoi na przeszkodzie, aby do jego zasilania zastosować zewnętrzne źródło napięcia 7...12 V. Pomocą przy montażu układu będą **fotografia 4** i **fotografia 5**.

Oprogramowanie mikrokontrolera

Program do miernika obrotów został napisany za pomocą Bascoma AVR, a jego działanie nie jest skomplikowane. Wykorzystuje on dwa timery mikrokontrolera. Timer0 (8-bitowy) wspomaga odczyt klawiatury i wyświetlanie na LCD. Jego przerwanie ustawia zmienną (flagę) co 4 ms. Timer1 (16-bitowy) odpowiada za generowanie przebiegu prostokątnego sterującego diodą LED. Procedurę obsługi przerwania od układu Timer1 zamieszczono na **listingu 1**.

Bezpośrednio po wywołaniu przerwania licznik w timerze przyjmuje wartość *D*.

Ta zmienna jest rezultatem obliczeń, które są wykonywane po każdorazowej zmianie generowanej przez procedurę zmiany częstotliwości błysków. Dalej jest zwiększana wartość zmiennej *Div* w przedziale od 0 do *Div_licz*. Jest to licznik programowego PWM, którego wartość końcowa zależy od zakresu pomiarowego. Wypełnienie PWM jest ustalone poprzez zmienną *Div_pulse*, która na jest bieżąco porównywana z wartością licznika. Rezultatem porównania jest zmiana poziomu logicznego na pinie dołączonym do bazy tranzystora sterującego diodą LED.



Rysunek 2. Schemat montażowy obrotomierza

Wykaz elementów

Rezystory: (SMD 0805)

- R1: 47 Ω
- R2: 3,3 kΩ
- R3: 330 Ω
- R4: 30 Ω/2 W (lub 4 szt. 120 Ω/0,5 W)
- P1: 10 kΩ potencjometr montażowy

Kondensatory:

- C1, C2: 22 pF (SMD 0805)
- C5...C7: 100 nF (SMD 0805)
- C3: 220 μF
- C4: 47 μF

Półprzewodniki:

- D1: dioda LED o mocy 0,5 W

T1: BC556

T2: BC337

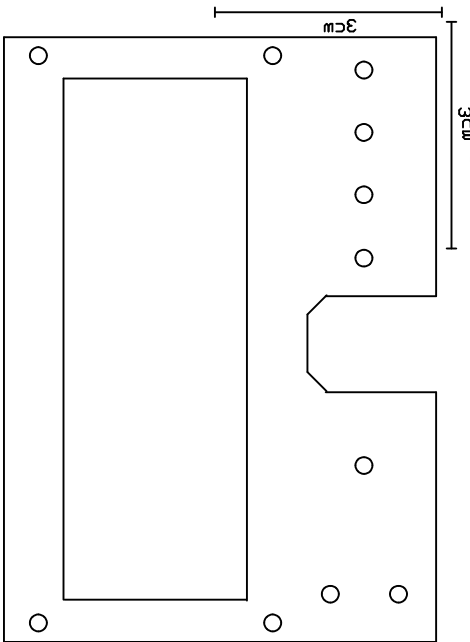
U1: ATmega8-16AU (lub ATmega8A)

U2: Stabilizator 78L05

Inne:

- Prog: goldpin kątowy 1×6
- S1...S6: mikroprzełącznik z oską 15 mm
- IMP: enkoder (impulsator) obrotowy
- X1: rezonator kwarcowy 16 MHz
- W1: wyświetlacz LCD 16×2 znaki
- Zasobnik na baterię 9 V/6F22





Rysunek 3. Plan wykonania otworów w obudowie KM35

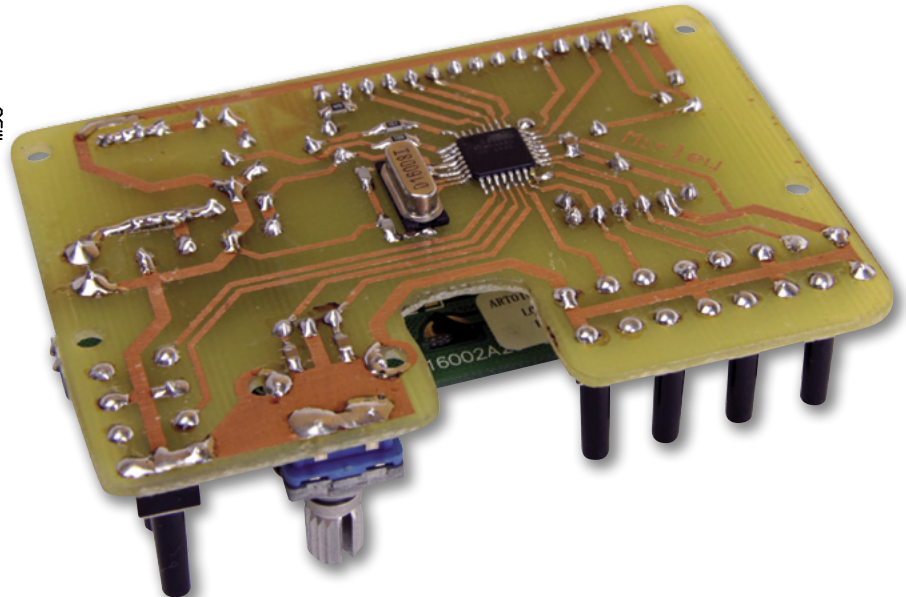
Procedura widoczna na **listingu 2** dba o to, aby do Timera1 została załadowana poprawna wartość początkowa.

Działanie procedury zależy od zmiennej *Obr*, która oznacza liczbę obrotów na minutę $\times 10$. Gdy $Obr \geq 4800$, to dzięki zmiennym *Div_licz*=32 i *Div_pulse*=1 na wyjściu pojawia się przebieg o wypełnieniu równym 1/32. W najprostszym wypadku wystarczyłoby przeliczyć $T=300000000/Obr$ i rzutować wartość T do zmiennej typu word, aby zapewnić generowanie odpowiedniej częstotliwości na wyjściu. Wartość 300000000 wynika z podzielenia 16 MHz zegara przez 32 kroki programowego PWM i pomnożeniu przez 600 obrotów/min (obroty są $\times 10$). Testy i pomiary wykazały, że częstotliwość nie zgadza się z teoretycznymi rozważaniami. Istotny wpływ na ten efekt ma obsługa przerwań i odkładanie na stos wartości rejestrów mikrokontrolera, dlatego wykorzystano kwadratową krzywą kalibracyjną pozwalającą wyliczyć wartość licznika Timer0 na podstawie ustawionych obrotów. Opis procedury kalibracji zostanie przedstawiony dalej.

Gdy obroty są mniejsze od wspomnianej wcześniej wartości 4800, to *Div_licz*=256 i *Div_pulse*=8. Wypełnienie przebiegu jest takie samo, ale częstotliwość zostaje podzielona wstępnie przez 8. Reszta obliczeń jest wykonywana analogicznie.

Na samym końcu wartość zmiennoprzecinkowa zamieniana jest na word i jest wyliczany błąd względny wyrażony w obrotach/min, który powstaje przez odrzucenie miejsc po przecinku. Błąd ten jest wyświetlany na LCD razem z ustawioną wartością obrotów.

Na **listingu 3** zaprezentowano fragment pętli głównej odpowiedzialny za obsługę impulsatora.



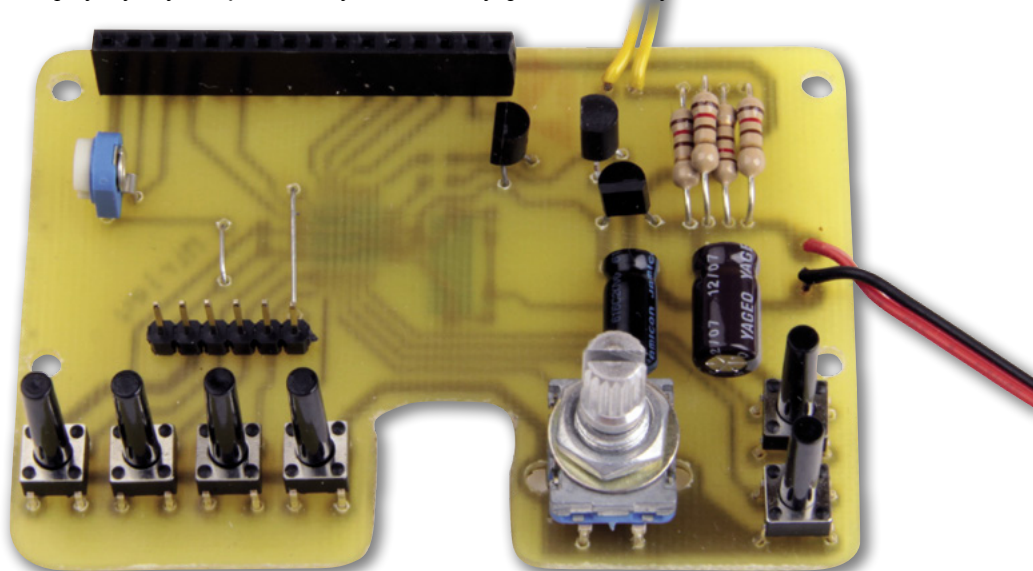
Fotografia 4. Zmontowany prototyp obrotomierza w widoku od spodu

Do zmiennych bitowych *In1* oraz *In2* są wpisywane stany doprowadzeń wejściowych, do których jest dołączony impulsator. Zmienna *Rezultat* przyjmuje wartość 0 na początku każdego wywołania procedury. Zmienna *Tryb* jest równa 0, a program czeka, aż oba piny impulsatora znajdują się na poziomie niskim lub wysokim. W następnym kroku zmienna *Tryb* przyjmie wartość 1 lub 2, dzięki czemu będą teraz sprawdzane dwa przypadki zmiany stanów logicznych, zależne od kierunku obrotów osi impulsatora. Następnie jest wykonywany skok do wartości *Tryb* równej 3 lub 4 i zostanie ustawiona zmienna *Rezultat*. Daje ona informacje o ukrytym kierunku ruchu impulsatora. Końcowa część procedury (niewidoczna na **listingu 3**) zajmuje się modyfikacją zmiennej *Obr* w zależności od kierunku kręcenia i wciśniętych przycisków mnożnika (S3...S6). Opisana procedura nie używa przerwań, czego rezultatem może być gubienie impulsów przy szybszym kręceniu. Pełny kod źród-

łowy zawiera w komentarzu drugą podobną procedurę odczytu impulsatora, która może się przydać, gdy zostanie zastosowany impulsator innego typu.

Na **listingu 4** pokazano fragment obsługi jednego z pozostałych dwóch przycisków (S1). Drugi działa w sposób analogiczny.

Procedura ta wykonywana jest co 4 ms dzięki przerwaniom generowanym przez Timer0. Na bieżąco jest sprawdzany stan pinu przycisku SW1 i jeśli zostanie on zwarty do masy, to nastąpi zliczanie impulsów w zmiennej *Licz_sw(1)*. Zapewnia ona eliminację drgań styków, gdyż reakcja na naciśnięcie pojawi się tylko wtedy, gdy przycisk jest trzymany przynajmniej przez 80 ms. Jeśli w międzyczasie pojawi się stan wysoki, zmienna *Licz_sw(1)* zostanie wyzerowana i nie nastąpi żadna reakcja. Prawidłowe naciśnięcie przycisku mnoży ustawione obroty przez 2, sprawdza, czy nie przekroczyły one maksymalnej wartości i wywołuje procedurę: przeliczania i wyświetlania.



Fotografia 5. Zmontowany prototyp obrotomierza w widoku od góry

Listing 1: Procedura obsługi przerwania od układu Timer1

```
Prztimer1:
  Timer1 = D <max 65535
  Incr Div
  If Div >= Div_licz Then Div = 0
  If Div < Div_pulse Then Led = 1 Else Led = 0
Return
```

Listing 2: Procedura obliczająca wartość początkową licznika Timer1

```
Przelicz:
  If Obr >= 4800 Then
    Div_licz = 32
    Div_pulse = 1
    T = 0.0000002546 * Obr
    T = T * Obr
    T2 = 0.9969052 * Obr
    T = T + T2
    T = T + 1.00037
    T = 300000000 / T
  Else
    Div_licz = 256
    Div_pulse = 8
    T = 0.0000018862 * Obr
    T = T * Obr
    T2 = 0.999905 * Obr
    T = T + T2
    T = T + 0.189869
    T = 37500000 / T
  End If

  X = T
  D = 65536 - X

  T2 = T - X
  T = T2 / T
  T = T * Obr
Return
```

Listing 3: Fragment procedury obsługi impulsatora

```
In1 = I1
In2 = I2
Rezultat = 0
Select Case Tryb
  Case 0:
    If In1 = 1 And In2 = 1 Then Tryb = 1
    If In1 = 0 And In2 = 0 Then Tryb = 2
  Case 1:
    If In1 = 1 And In2 = 0 Then Tryb = 3
    If In1 = 0 And In2 = 1 Then Tryb = 4
  Case 2:
    If In1 = 0 And In2 = 1 Then Tryb = 3
    If In1 = 1 And In2 = 0 Then Tryb = 4
  Case 3:
    If In1 = In2 Then
      Rezultat = 1
      Tryb = 0
    End If
  Case 4:
    If In1 = In2 Then
      Rezultat = -1
      Tryb = 0
    End If
End Select
```

Listing 4: Fragment procedury obsługi przycisku S1

```
If Sw1 = 0 Then
  Incr Licz_sw(1)
  If Licz_sw(1) = 20 Then
    Obr = Obr * 2
    If Obr > 420000 Then Obr = 420000
    Gosub Przelicz
    Gosub Wyswlcid
  End If
Else
  Licz_sw(1) = 0
End If
```

Listing 5: Fragment procedury obsługi LCD

```
Locate 1 , 2
If Obr < 100000 Then Lcd " "
If Obr < 10000 Then Lcd " "
If Obr < 1000 Then Lcd " "
X = Obr / 10
Lcd X ; "."
X = Obr Mod 10
Lcd X
Locate 1 , 10
Lcd Chr(0) ; «n/min» ; Chr(1)
```

BSTDISBL	<input type="checkbox"/>	Select if PC6 is I/O pin or RESET pin
WDTON	<input type="checkbox"/>	Watchdog timer always on
SPEN	<input checked="" type="checkbox"/>	Enable Serial Program and Data Downloading
CKOPT	<input checked="" type="checkbox"/>	Oscillator options
EESAVE	<input type="checkbox"/>	EEPROM memory is preserved through the Chip Erase
BOOTSZ1	<input checked="" type="checkbox"/>	Select Boot Size (see Table 82 for details)
BOOTSZ0	<input checked="" type="checkbox"/>	Select Boot Size (see Table 82 for details)
BOOTRST	<input type="checkbox"/>	Select Reset Vector
BODLEVEL	<input type="checkbox"/>	Brown out detector trigger level
BODEN	<input type="checkbox"/>	Brown out detector enable
SUT1	<input type="checkbox"/>	Select start-up time
SUT0	<input type="checkbox"/>	Select start-up time
CKSEL3	<input type="checkbox"/>	Select Clock source
CKSEL2	<input type="checkbox"/>	Select Clock source
CKSEL1	<input type="checkbox"/>	Select Clock source
CKSEL0	<input type="checkbox"/>	Select Clock source

Rysunek 6. Ustawienia fusebitów mikrokontrolera ATmega8

Listing 5 przedstawia fragment procedury wyświetlania obrotów na LCD.

Jeżeli wartość obrotów jest mniejsza od 100000, 10000 lub 1000, to zostają wyświetlone dodatkowe spacje, aby zachować wyrównanie liczby do prawej strony. Ponieważ obroty reprezentowane przez zmienną pomnożone są przez 10, następuje dzielenie zmiennej na część dziesiątek i jedności. Na samym końcu zostaje wyświetlona jednostka „[n/min]”. W drugiej linii wyświetlany jest błąd na dokładnie takiej samej zasadzie jak opisane wyżej obroty.

Oprócz zaprogramowania mikrokontrolera ważne jest ustawienie jego fusebitów. Mikrokontroler powinien pracować z zewnętrznym rezonatorem kwarcowym 16 MHz, zatem CKSEL=1111, SUT=11 oraz CKOPT=0. Ustawienie fusebitów ilustruje rysunek 6.

Kalibracja i pomiary

Po zaprogramowaniu układu wstępną wersją programu i wylczeniu podzielnika timera, pomiary wykazały odchyłki generowanej częstotliwości w stosunku do spodziewanej teoretycznie. Jest to spowodowane pracą z podzielnikiem timera równym 1, czasem jaki jest potrzebny na obsługę przerwania i czasem przeładowania wartości w rejestrze timera. W tabeli 1 podano pomiary częstotliwości (kilka przykładowych wartości) generowanej na wyjściu (f_p) w stosunku do częstotliwości, jaka powinna być dostępna teoretycznie (f_i) oraz odpowiadające im wartości obrotów (obroty pomnożone są przez, 10 aby uzyskać dokładność nastaw 0,1 obrotów/min).

Na podstawie danych z tabeli 1 zostały sporządzone wykresy zależności obrotów teoretycznych i mierzonych w dwóch przedziałach: 60...480 obr./min i 480...42000 obr./min. Podział ten wynika z funkcjonowania programu, w którym są dwa zakresy pomiarowe. Na rysunku 7 pokazano punkty pomiarowe wraz z dopasowaną krzywą kalibracyjną dla pierwszego zakresu pomiarowego. Dla drugiego wykres wygląda bardzo podobnie.

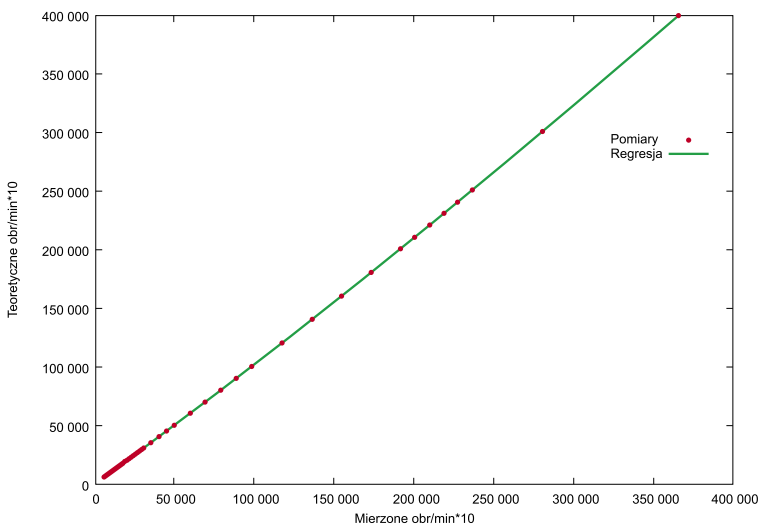
Dane z obu wykresów dobrze pasują do zależności kwadratowej, przy czym współczynniki przy drugich potęgach są bardzo małe w obydwu przypadkach (zależność „prawie” liniowa). Dopasowanie krzywej drugiego stopnia zostało wykonane w programie Gnuplot (<http://www.gnuplot.info>).

Tabela 1: Pomiary częstotliwości przed kalibracją

f_p [Hz]	f_i [Hz]	n_p [obr/min]	n_i [obr/min]
0,999	1,000	599	600
2,493	2,500	1496	1500
16,624	16,667	9974	10000
18.283	18,333	10970	11000
82,382	83,333	49429	50000
98,638	100,000	59183	60000
468,080	500,000	280848	300000
611,135	666,667	366681	400000

Tabela 2: Pomiary częstotliwości po kalibracji

Obr/min	f_{pom} [Hz]	f_{ideal} [Hz]	Błąd [%]	Błąd [Obr]
60	0,99997	1,000	-0,003	-0,0018
120	1,97900	2,000	-1,050	-1,26
3000	49,950	50,000	-0,100	-3,00
6000	99,860	100,000	-0,140	-8,40
36000	599,010	600,000	-0,165	-59,4
42000	698,580	700,000	-0,203	-85,2



Rysunek 7. Wykres kalibracji miernika na zakresie 480...42000 obr./min

Wprowadzone krzywe kalibracyjne w dwóch zakresach pomiarowych znacznie polepszyły dokładność miernika, o czym można się przekonać, patrząc na przykładowe pomiary zamieszczone w tabeli 2. Teraz błąd generowanej częstotliwości jest tylko ułamkiem procenta wartości ustawionej, a urządzenie śmiało można nazwać obrotomierzem, a nie tylko wskaźnikiem obrotów.

Zasada pomiaru

Gdy wirujący przedmiot jest oświetlony światłem ciągłym, widać rozmazany obraz, który nie niesie ze sobą żadnej informacji o badanym obiekcie. Całkiem inaczej jest, jeśli ten sam obiekt zostanie oświetlony światłem impulsowym, a ściślej krótkimi błyskami świetlnymi. Jeśli wspomniane błyski będą zawsze przypadać na tę samą fazę ruchu obrotowego (częstotliwość błysków równa częstotliwości wirowania), to uzyskany obraz pozorny będzie nieruchomy (zjawisko stroboskopowe). Jeżeli obie częstotliwości nie są jednakowe, a tylko zbliżone do siebie, to wynikiem obserwa-

cji będzie powolny ruch obiektu zgodnie lub przeciwnie do kierunku ruchu rzeczywistego (zależnie od relacji częstotliwości obrotów i częstotliwości błysków).

W celu łatwego przeprowadzenia pomiaru należy oznaczyć wirujący element kreską idącą wzdłuż promienia. Po wprawieniu urządzenia w ruch znacznik nie będzie widoczny, dopóki nie zostanie oświetlony światłem stroboskopowym. Należy tak ustawić częstotliwość błysków, aby uzyskać pojedynczy, nieruchomy obraz narysowanej uprzednio kreski. Następnie wykorzystując przycisk $\times 2$ można podwoić częstotliwość, aby upewnić się, czy na dwukrotnie większej też nie ma przypadkiem pojedynczego obrazu. Częstotliwość wirowania to maksymalna częstotliwość, przy której widoczny jest pojedynczy obraz. Jest to tylko jeden przypadek, jaki można spotkać podczas pomiaru, opis wszystkich szczegółów podczas mierzenia tą metodą znacznie wykracza poza łamy tego artykułu.

Mirosław Firlej
elektronika@firlej.org