



Obrotomierz stroboskopowy

kit

3136

AVT

część 1

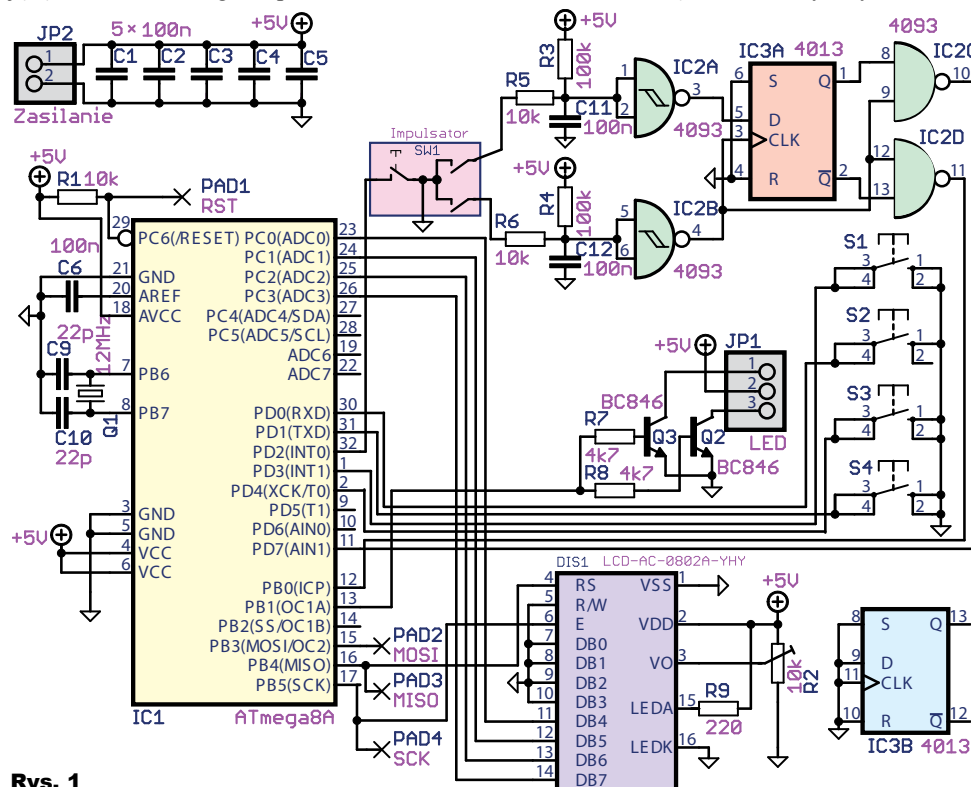
W pełni bezinwazyjny miernik prędkości obrotowej z automatycznym obliczaniem na podstawie metody nieruchomych obrazów. Poręczny i wygodny, przyda się w każdym warsztacie!

Zgodnie z nazwą, urządzenie mierzy prędkość obrotową wirników. Wykorzystuje do tego metodę stroboskopową, polegającą na oświetleniu go impulsami światła

o regulowanej częstotliwości i uzyskaniu złudzenia nieruchomych obrazów. Jest to metoda całkowicie bezinwazyjna, dająca też interesujące efekty wizualne. Można powiedzieć, że jest to obrotomierz półautomatyczny. Ta „półautomatyczność” polega na zaimplementowaniu funkcji obliczania faktycznej prędkości obrotowej na podstawie wskazania częstotliwości dwóch sąsiednich statycznych

obrazów. Urządzenie jest w stanie zrealizować błyski diod o częstotliwości 300 do 10000 błysków/min, a implementacja metody obrazów pociągnęła za sobą wielką korzyść – umożliwia pomiar wirników o prędkości znacznie większej niż wynosi górna granica tegoż zakresu. Nie było dla tego urządzenia problemem zmierzenie obrotów wentylatora, wynoszących ok. 22000 obr/min z przyzwolitą dokładnością. Po wykonaniu takiego „nadmiarowego” (i zwykłego też rzecz jasna!) pomiaru obrotomierz umożliwia natychmiastową weryfikację poprawności, wytwarzając błyski o różnych częstotliwościach podwielokrotnych.

Całość zamknąłem w wygodnej, podłużnej obudowie w kształcie pilota od TV, wyposażylem w czytelny, alfanumeryczny wyświetlacz LCD z podświetleniem, cztery przyciski microswitch oraz obrotowy impulsator z przyciskiem. Do wytwarzania błysków służy 10 białych diod LED o dużej jasności, umieszczonych pod dwoma różnymi kątami (dla większej wygody obsługi). Dalsze techniczne szczegóły przedstawię w artykule, a tutaj dodam jeszcze, że urządzenie może być też bardzo interesującą niecodzienną zabawką – możemy w świetle stroboskopu zobaczyć w zwolnionym tempie szybkie drgania różnych przedmiotów. Szczególnie polecam obejrzeć efekty „zamrożenia” w snopie światła drgającej struny jakiegoś instrumentu!



Rys. 1

Opis układu

Schemat elektryczny obrotomierza jest pokazany na **rysunku 1**. Głównym elementem urządzenia jest mikrokontroler ATmega8A. Jest on taktowany rezonatorem kwarcowym o częstotliwości 12MHz. Im wyższa jest ta częstotliwość, tym dokładniejszy będzie miernik, pobór prądu wprawdzie przy tym wzrośnie, ale to nie jest problemem. Problem tkwi w obszarze *Safe Operating Area* mikrokontrolera (stosowny rysunek można znaleźć w danych katalogowych elementu), czyli wymagania co do napięcia zasilania w zależności od częstotliwości zegara. Chcąc zastosować kwarc 16MHz, musiałbym zapewnić napięcie nie mniejsze niż 4,5V. Do obudowy zmieszczą się 3 baterie AAA zasilające układ (na schemacie zasilanie z nich oznaczone jest +5V, co jest przybliżeniem). Przy nowych bateriach nie ma problemu, ale wystarczy, że nieco się rozładują i zostaną obciążone, a napięcie „przysiadzie” i praca mikrokontrolera zacznie być niestabilna. Do pracy z zegarem 12MHz, jak można oszacować z wykresu w specyfikacji, wystarczy zasilanie 3,6V. Taka częstotliwość jest też wystarczająca do sprawnej pracy urządzenia bez „lagów” (opóźnień, które miały miejsce przy taktowaniu 8MHz) podczas szybkiego obracania impulsatora. Najistotniejszym zadaniem ATmega8A jest generowanie krótkich impulsów (błyskanie diodami) o dokładnie ustalonej, regulowanej z możliwie małym krokiem częstotliwości. Procesor ma jeszcze do obsługi wyświetlacz LCD, a całość musi pracować sprawnie i szybko reagować na polecenia użytkownika (zwłaszcza szybkie obracanie oską impulsatora), aby obsługa była łatwa i komfortowa.

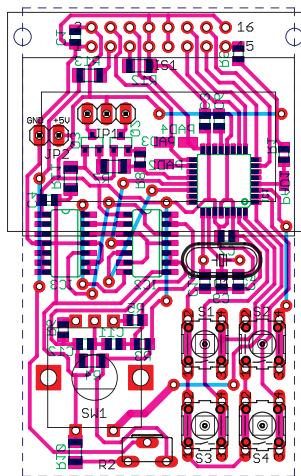
Do mikrokontrolera podłączone są przyciski i alfanumeryczny wyświetlacz LCD 8x2 z podświetleniem włączonym na stałe (o jasności decyduje wartość R9). Kontrast regulujemy potencjometrem R2. Przyciski, łącznie z tym w impulsatorze, zwierają do masy, a podciąganie z drugiej strony do plusa zrealizowałem, włączając wewnętrzne rezystory pull-up do tych wyprowadzeń mikrokontrolera. Obwody z tranzystorami Q2 i Q3 sterują razem diodami LED, w moim układzie każdy z tych tranzystorów ma pod sobą pięć diod. Rezystory ograniczające prąd włączamy poza płytką, możemy więc łatwo eksperymentować z liczbą LED-ów.

Chyba najciekawszym fragmentem schematu jest blok za wyjściami „obrotowymi” impulsatora, zawierający kostki IC2, IC3. Już na etapie projektowania hardware’u obrotomierza wiedziałem, że mikrokontroler i tak będzie już „bardzo

zajęty” innymi zadaniami, nie chciałem jeszcze obciążać go obsługą impulsatora i dodatkową eliminacją drgań jego styków – w przeciwieństwie do przycisków microswitch, impulsator będzie wytwarzał o wiele szybsze przebiegi do obsługi, a jak wspominałem – dla mnie opóźnienia („lagi”) są nie do przyjęcia. Dlatego postanowiłem zbudować sprzętowy dekoder do impulsatora, z wbudowaną eliminacją drgań styków. Na jednym ze swoich dwóch wyjść, podłączonych do mikrokontrolera, wystawia tylko pojedynczy impuls – na pierwszym, gdy przekroczymy oskę zgodnie z ruchem wskazówek zegara (CW) o pojedynczy krok, a na drugim, gdy przeciwnie (CCW).

Omówienie zaczęę od tego, jak zachowuje się impulsator podczas pracy: przy pojedynczym kroku (przeważnie 1/12 lub 1/24 pełnego obrotu w zależności od modelu) następuje chwilowe zwarcie obu wyjść do masy (dla normalnie rozwartego – są wersje odwrotne, rozwierające). Te zwarcia następują jednak z pewnym przesunięciem fazowym (przeważnie niewielkim), tzn. jedno z wyjść zwierza wcześniej niż drugie – które to zależy od kierunku obrotu. Jest to mechanicznie łatwe do zrealizowania. Rozwieranie również następuje w takiej samej kolejności. Najpierw następuje eliminacja drgań styków – producent w specyfikacji zaleca, by opóźnienie wyniosło ok. 5ms. Zostało to zrealizowane na układach RC oraz inwerterach z histerezą. Zwróćmy uwagę na rezystory R5 i R6 – ograniczają one impuls prądowy, płynący przez styki impulsatora, podczas rozładowywania kondensatora. Jest mnóstwo projektów, w których w układach eliminacji drgań również stosuje się układy RC, ale styki impulsatora wstawia się równolegle do kondensatora. W momencie zwarcia tych styków płynie bardzo duży prąd chwilowy, który nie tylko drastycznie przyspiesza zużycie impulsatora, ale również sieje ogromną ilość zakłóceń w bardzo szerokim spektrum. Mając tak zestawiony układ, można te zakłócenia usłyszeć, ustawiając w pobliżu radio AM (jest dużo filmików na YouTube pokazujących to zjawisko). Obecność rezystorów R5, R6 praktycznie eliminuje tę wadę.

Za inwerterami (zrobionymi z bramek NAND) znajduje się przerzutnik typu D na układzie IC1A (połowa 4013, druga



Rys. 2

rzutnikiem znajdując się pozostałe bramki NAND układu IC2, które „przepuszczają” zawartość oryginalną (Q) lub zanegowaną (/Q) w momencie, kiedy na wyjściu „zegarowym” jest jedynka. W ten sposób na wyjściu odpowiedniej bramki pojawia się impuls (logicznego zera) w momencie, kiedy obrócimy oskę impulsatora, a gdy stoi w miejscu – na obu jest logiczna jedynka, czyli „nic się nie dzieje”. Dzięki tak prostej konstrukcji obwód nie „zatnie się”, kiedy impulsator wygeneruje jakiś nieprzewidziany sygnał, który przedstanie się przez eliminatory drgań. Może się to zdarzyć wskutek wibracji, wad lub zatarzenia się impulsatora. Z punktu widzenia programowania ATmega8A, traktujemy wyjścia z tego dekodera jak dwa przyciski, dodatkowo wolne od drgań styków. Myślę, że przedstawiony tutaj układ obsługi impulsatora zwrócił na siebie uwagę tych Czytelników, którzy zechcą użyć go w niektórych swoich projektach, kiedy (tak jak tu) w uzasadniony sposób będzie trzeba odciać mikrokontroler lub w ogóle nie będzie w projekcie układu programowalnego.

Projekt obrotomierza został maksymalnie uproszczony, aby zmieścić wszystkie elementy na niewielkiej płytce drukowanej (w małej obudowie, oprócz diod i LCD, muszą jeszcze zmieścić się trzy baterie!). Druga sprawa to zapewnienie dużej elastyczności w przygotowywaniu i modyfikowaniu oprogramowania. W dalszej części artykułu omówię pewne aspekty programu oraz metody wytwarzania i regulacji częstotliwości, aby każdy mógł poznać moje sposoby na to, a także wprowadzić zmiany w programie.

Montaż i uruchomienie

Zaprojektowałem jednostronną płytkę drukowaną, aby łatwo było można ją wykonać w domu metodą termotransferu. Projekt można zobaczyć na **rysunku 2**. Jest ona tak duża, na ile pozwalała obudowa Z-14. Elektronika jest dość gęsto

Fot. 1



Fot. 2

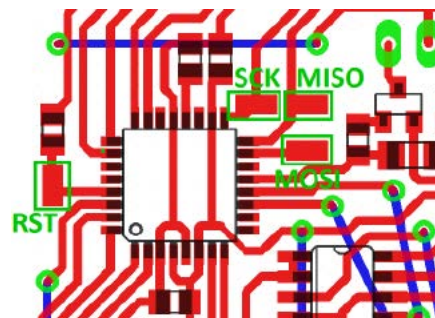


upakowana, ale wbrew pozorom montaż nie jest specjalnie trudny. Ktoś, kto choć raz lutował SMD z układami scalonymi, bez problemu da sobie radę, jeśli tylko nie skończył mu się topnik. Zaczynamy właśnie od układów scalonych, warto do nich użyć grota minifala. Następnie montujemy wszystkie pozostałe elementy powierzchniowe, kończąc na tranzystorach. Pamiętajmy o zworach SMD 1206, których nie zaznaczyłem na schemacie ideowym. Po sprawdzeniu, czy nie zrobiliśmy gdzieś zwarć, przechodzimy do elementów przewlekanych, zaczynamy od zworek z drutu. Po nich rezonator kwarcowy i potencjometr do regulacji kontrastu wyświetlacza. Teraz najlepiej zamontować przewody – zasilania oraz diod. Anody diod łączymy do środkowego pada JP1. Przewody powinny mieć ok. 10–15cm długości, później je skrócimy. Po kablach lutujemy cztery przyciski mikroswitch o długości osi 13,5mm, a po nich element najwyższy – impulsator ED1212S o długości osi 30mm. Ostatni element do lutowania na płytce drukowanej to wyświetlacz LCD – ja najpierw przyluto-

wałem komplet goldpinów na PCB, a następnie dobrałem wysokość montażu wyświetlacza, kiedy już przygotowałem do tego obudowę. Otwory na przyciski, impulsator i wyświetlacz wykonałem, drukując odpowiednie warstwy projektu z programu Eagle w odbiciu lustrzanym, przykleiłem od wewnątrz obudowy i zaznaczyłem, gdzie wiercić i ciąć. Włożyłem płytkę w obudowę i szukałem odpowiedniej wysokości umieszczenia LCD, podkładając pod niego zwinięty papier. Jak już ustawiłem odpowiednią wysokość, ostrożnie przylutowałem wyświetlacz do płytki. **Fotografia 1** pokazuje mój model od środka.

Teraz pora zaprogramować mikrokontroler. Na **rysunku 3** oznaczone są pady na płytce, umożliwiające wygodne przylutowanie przewodów programatora ISP. Pamiętajmy o podłączeniu zasilania do JP2, zwracając uwagę na poprawność polaryzacji. Oprócz załadowania programu (listing i skompilowane wsady do pobrania w Elportalu wśród materiałów dodatkowych), trzeba jeszcze zmienić ustawienia fusebitów: ustawić jako źródło taktowania zewnętrzny rezonator kwarcowy o dużej częstotliwości, a także odblokować BOD (Brown-Out Detector) i ustawić na 2,7V, dzięki czemu mikrokontroler będzie sam się resetował, kiedy napięcie zasilania opadnie poniżej tej wartości i nie ma ryzyka jego zawieszenia przy włączaniu zasilania.

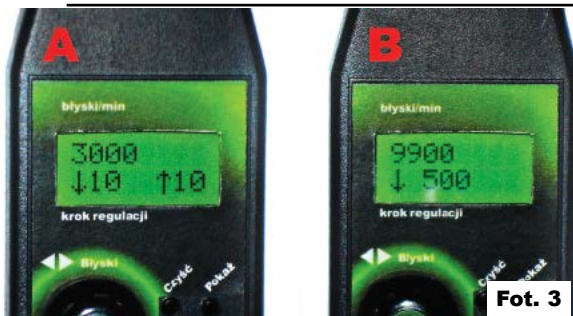
Po zaprogramowaniu i zresetowaniu układu powinien on zacząć dzia-



Rys. 3

łać. Zanim będzie w pełni użytecznym urządzeniem, trzeba oczywiście zamontować diody oraz koszyk na baterie zasilające. Przypomnę także o ustawieniu potencjometrem kontrastu wyświetlacza LCD. Miejsce koszyka jest oczywiście z tyłu obudowy, trzeba tylko ściąć jedną ściankę korytka (dostosowanego do dwóch baterii). Lutujemy do niego przewody zasilające i podłączamy elektronikę, zostawiając „po drodze” miejsce na wyłącznik zasilania. Wkładamy do koszyka baterie (aby osiągnął „robocze” gabaryty) i przyklejamy go klejem z pistoletu do obudowy. Przełącznik zasilania stosujemy dowolny, ja użyłem małego dźwigniowego, bo łatwo się go montuje – widać go na fotografiach. Wadą jego jest to, że łatwo go niechcący przełączyć (np. podczas transportu). Pewnym utrudnieniem jest też to, że w obudowie jest mało miejsca. Mogę polecić montaż przełącznika właśnie w tym miejscu, co w moim obrotomierzu, ponieważ nie przeszkadza on w obsłudze jedną dłonią. Jeśli chodzi o diody stroboskopowe, mamy tutaj dowolność co do ich liczby oraz typu, byle tylko się pomieściły. U siebie zastosowałem 10 jasnych diod LED 3mm, pięć sztuk na frontowej ściance (jak w pilocie od TV), a drugie pięć na ściance ukośnej poniżej – **fotografia 2**. Takie rozwiązanie umożliwia wygodne oświetlenie wirników różnej konstrukcji z widokiem na wyświetlacz, bez zmuszania nas do gimnastyki nadgarstka. Diody są umieszczone ciasno, anody wszystkich zostały zwarte i podłączone do środkowego pada JP1. Do katody każdej z nich przylutowałem po rezystorze 33Ω, których przeciwne końcówki również pozwierałem, ale po 5 diod, po czym każdą taką piątkę katod podłączyłem do pozostałych padów JP1. Lutując diody, uważamy, aby ich nie

R E K L A M A



Fot. 3

przebrać (kiedyś mi się zdarzyło). Zastosowanie dwóch osobnych tranzystorów odciąża każdy z nich przy sterowaniu dużymi prądami i tym samym zwiększa niezawodność urządzenia przy większej liczbie diod. LED-y mocujemy klejem z pistoletu, przy czym nie możemy zalać rezystorów, bo przy dłuższej pracy klej może zacząć się topić. Nie bójmy się małych rezystancji ograniczających prądy diod, one pracują impulsowo i nie ma co im żałować prądu!

Obrotomierz jest prawie gotowy. Proponuję nakleić warstwę opisów na górną powierzchnię obudowy. Naklejka (kolorowa i czarno-biała) jest dostępna wśród materiałów dodatkowych w Elportalu, w różnych formatach, usunąłem też swoje imię i nazwisko oraz oznaczenie wyłącznika zasilania. Ja wydrukowałem etykietę na papierze samoprzylepnym do drukarki, a pod przyciski wyciąłem prostokąty – jest to rozwiązanie najprostsze, choć mniej eleganckie. Na naklejkach są krzyżyki, przez które można naciskać przyciski jak w klawiaturze membranowej. Końcowy etap to umocowanie wszystkiego w obudowie – jeśli prawidłowo dobraliśmy wysokość montażu LCD, wystarczy przykleić płytkę na brzegach klejem z pistoletu do dolnej części obudowy. Wkładamy baterie, skręcamy obudowę i zakładamy gałkę na impulsator. Polecam taką, którą wykorzystałem w swoim modelu (gałkę można kupić w AVT, podobnie jak wszystkie inne detale wymienione w wykazie elementów), ponieważ ma odpowiednie wymiary, kolorem pasuje do LCD i da się łatwo zdrapać białą kreskę (wskaźnik), która przy impulsatorze nie ma żadnego sensu. Gałkę przykręcamy śrubokrętem (zostawiamy zapas wysokości, aby dało się ją nacisnąć).

W tym momencie budowa obrotomierza została ukończona. Warto się do niej przyłożyć na każdym etapie, bo urządzenie może być zmuszone do pracy w niełatwych warunkach, często też, jak to bywa w warsztacie, narażone jest na upadki i inne przeciążenia mechaniczne. Warto na wyświetlaczu pozostawić folię lub nakleić nową (do nabycia w sklepach z akcesoriami GSM), zwłaszcza na brzegach jego części roboczej, aby nie wchodziły do środka pod jego konstrukcję pyłki i zanieczyszczenia.

Staralem się, aby urządzenie było maksymalnie proste w obsłudze i intuicyjne, a pomiary mogły być wykonywane możliwie szybko, sprawnie i wymagające jak najmniej klikania. Po włączeniu urządzenia, na górnej linijce wyświetlacza zobaczymy bieżącą częstotliwość błysków diod, wynoszącą na starcie 3000 błysków/min. Na dolnej linijce widoczny jest krok regulacji. Domyślnie ustawiony jest tzw. krok minimalny, czyli najmniejszy, który gwarantuje nam mały błąd (odchyłki) generowanej częstotliwości od założonej, o czym opowiem szczegółowo dalej. Krok minimalny rośnie wraz z ustawioną częstotliwością w sposób skokowy (wynosi 5, 10, 25 lub 50). Jeśli jest on wybrany, dolna linijka wygląda jak na **fotografii 3a**, ukazuje ona wyświetlacz obrotomierza zaraz po włączeniu zasilania. Kręcąc gałką impulsatora, zmieniamy częstotliwość o wartość kroku widocznego na wyświetlaczu. Naciskając na gałkę, zmieniamy krok. Do wyboru są: minimalny (zależny od częstotliwości), 50, 500, 2x (mnożenie lub dzielenie przez 2). Jest to wygodne rozwiązanie do szukania statycznych obrazów wirnika. W tym podrozdziale przedstawię jeszcze funkcję przycisku *Przybliż do 500* – jak nazwa wskazuje, zaokrągla on ustawioną wartość częstotliwości do najbliższej wielokrotności 500 obr/min, ustawia też krok regulacji na 500. Znaczna większość silników elektrycznych ma zadeklarowaną prędkość obrotową, podawaną często w wielokrotności 1000 obr/min. Możemy zatem łatwo skontrolować, na ile jest to prawdą, mając za punkt wyjścia obietnicę producenta, kiedy na wyświetlaczu mamy ustawioną jakąś nieokrągłą liczbę obr/min. Na **fotografii 3b** pokazana jest interesująca sytuacja – kiedy wybrany jest krok 500 i ustawiona prawie maksymalna częstotliwość błysków diod – dlatego zniknęła strzałka w górę obok strzałki w dół (bo dodanie 500 przekroczyłoby

Wykaz elementów

R1, R5, R6	10kΩ SMD 0805
R2	potencjometr montażowy pionowy 10kΩ
R3	100kΩ SMD 0805
R4	100kΩ SMD 1206
R7	4,7kΩ SMD 1206
R8	4,7kΩ SMD 0805
R9	220Ω SMD 0805
R10-R13	„zwora” 0Ω SMD 1206 (nie ma na schemacie ideowym!)
C1-C6, C11, C12	100nF SMD 0805
C9, C10	22pF SMD 0805
Q2, Q3	BC846 lub podobny
IC1	ATmega8A TQFP32
IC2	4093 SMD
IC3	4013 SMD
DIS1	wyświetlacz LCD 2x8 (AVT: LCD2x8 01, inny nie pasuje!)
Q1	rezonator kwarcowy 12MHz niski
S1-S4	przycisk microswitch, ośka 13,5mm
SW1	impulsator z wyłącznikiem ED1212S ośka 30mm (AVT: IMPULSATOR9)
	Goldpiny proste męskie 2x8 (do mocowania LCD)

Elementy dodatkowe:

Diody białe 3mm ultrajazne	– 10 szt.
Rezystory 33Ω przewlekane 1/4W	– 10 szt.
Obudowa Z-14	– 1 szt.
Koszyk na baterie 3xAAA poziomy z oczkami lutowniczymi (AVT: KOSZYK BAT11)	– 1 szt.
Przełącznik dźwigniowy mały (AVT: SMTS102)	– 1 szt.
Gałka (GAŁ G19-GN)	– 1 szt.

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-3136.

10000 błysków/min). Krok minimalny jest na wyświetlaczu oznaczony dwiema osobnymi liczbami (w górę i w dół), nie tylko po to, aby się wyróżniał, ale też dlatego, że mogą one być różne – w sytuacji, kiedy stoimy na granicy dwóch jego wartości. Granice kroków minimalnych wynoszą 300 (dolna granica), 1400, 4500, 7000, 10 000 (górną).

Podane informacje wystarczą do wykonania przyrządu i przeprowadzenia podstawowych pomiarów. A w drugiej części artykułu przedstawione zostaną dodatkowe informacje na temat wykonywania pomiarów i ich specyfiki.



Michał Pędzimaz
mpedzimaz@gmail.com

R E K L A M A



Obrotomierz stroboskopowy

W pierwszej części artykułu podane były informacje dotyczące budowy i obsługi obrotomierza. Z uwagi na specyficzny sposób pomiaru i występujące przy tym pułapki, kwestię pomiarów trzeba omówić dokładnie.

Pomiary

Przedstawię teraz prawidłowy sposób wykonania pomiaru oraz przeznaczenie pozostałych przycisków. Dalsza część opisu, zawierająca wzory, niektórym Czytelnikom może nieco „namieszać w głowie”, jednak gdy się ma przyrząd w rękach i gdy przeprowadzi się kilka prób, wszystko powinno stać się jasne.

Generalnie naszym celem jest dążenie do ustawienia obrotowym impulsatorem takiej częstotliwości błysków, aby po oświetleniu diodami rotora, zobaczyć złudzenie, jakby stanął on w miejscu lub możliwie jak najwolniej się obracał. Efekty są naprawdę ciekawe, ale pamiętajmy, że to tylko niedoskonałość wzroku! Tak naprawdę on cały czas szybko wiruje i nie wolno zbliżać do niego rąk, a taka pokusa może się pojawić w przypadku „sztucznego” obrazu stroboskopowego. Jeśli dobraliśmy dokładnie taką samą częstotliwość błysków jak obrotów, wtedy odczytana z wyświetlacza wartość będzie właśnie liczbą obrotów na minutę badanego wirnika.

Ustawianie obrazu zaczynamy od największych częstotliwości z krokiem 500,

a kiedy zaczniemy odnosić wrażenie, że obraz rotora zwalnia, zmniejszamy kroki i dążymy do jak najmniejszych (najlepiej zerowych) obrotów obrazu, w praktyce uzyskuje się bardzo leniwie kręcący się wirnik, więc jest to proste i przyjemne. Musimy wiedzieć, jak wygląda wirnik, gdy stoi w miejscu (np. policzyć łopaty wentylatora), a później uzyskać dokładnie taki sam obraz. Może się zdarzyć, że będziemy błyskać co np. pół lub półtora obrotu, wtedy również obraz będzie nieruchomy, lecz „zlane” na nim dwa wentylatory – musimy na to uważać, nie brać „pierwszego lepszego” stojącego obrazu. To jest dość znana sprawa. Ale są też inne haczyki...

Fabryczne tachometry optyczne mają swoje zalety, ale są też w pewien sposób inwazyjne – należy na rotor nakleić niewielki kawałek odbijającej taśmy samoprzylepnej, a urządzenie ustawić bardzo blisko wirującej części (co może być niebezpieczne). Przedstawiony przeze mnie miernik zasadniczo jest tych wad pozbawiony, z małym wyjątkiem: problem może się pojawić, gdy rotor jest środkowosymetryczny, tzn. wygląda tak samo, gdy obróci się o kąt 180°. Wówczas musimy w jakiś sposób oznaczyć np. jedną z łopat wirnika, chociażby flamastrem. Przykładem takiego wirnika może być zacisk wiertarki lub ostrze blendera kuchennego. Wtedy nie unikniemy tego rodzaju inwazyjności.

Dруга, równie istotna sprawa, to podwielokrotności częstotliwości błysków. Kiedy już uzyskamy pożądany obraz, to skąd mamy wiedzieć, że błyskamy tak samo często, jak obraca się wirnik? Równie dobrze możemy błyskać rzadziej: co dwa, trzy, pięć obrotów... Aby nie zmuszać użytkownika do mozolnego poszukiwania, obliczania i innej matematyki, w urządzeniu znalazła się funkcja automatycznego wyznaczania szybkości obrotowej (właściwie półautomatyczna, bo użytkownik musi sam wskazać nieruchome obrazy). Metoda ta polega na wskazaniu dwóch **śsiednich** nieruchomych obrazów (tzn. częstotliwości błysków, przy której je uzyskujemy). Na ich podstawie obliczona zostanie prawdziwa liczba obrotów na minutę. Jak takie coś policzyć? Nic trudnego. Mamy dwa sąsiednie obrazy: występujące dla znalezionych częstotliwości błysków (ozn. X):

$$X1 = \frac{\text{obroty}}{a} \text{ oraz } X2 = \frac{\text{obroty}}{a+1}$$

Liczba $a = 1, 2, 3, \dots$, bo pierwszy obraz uzyskaliśmy dla częstotliwości błysków np. 3 razy mniejszej niż rzeczywiste obroty (liczba poszukiwana), a drugi – dla 4 razy mniejszej. Wtedy, rzecz oczywista, te obrazy są „sąsiedzi”. Z pierwszego wyznaczamy a :

$$a = \frac{\text{obroty}}{X1} \text{ i wstawiamy do drugiego równania:}$$

$$X2 = \frac{\text{obroty}}{1 + \frac{\text{obroty}}{X1}}$$

Znamy $X1$ i $X2$, bo te częstotliwości wskazywały urządzeniu jako wiarygodne, sąsiednie obrazy. Możemy wyznaczyć z tego równania poszukiwaną zmienną:

$$\text{obroty} = \frac{X2}{1 - \frac{X2}{X1}} = \frac{X1 * X2}{X1 - X2}$$

Ten wzór należy jeszcze nieco zmodyfikować:

$$\text{obroty} = \frac{X1 * X2}{|X1 - X2|}$$

wówczas będzie wszystko jedno, czy pierwsza częstotliwość jest większa, czy mniejsza od drugiej. To działanie matematyczne wykonuje procesor miernika, wyręczając użytkownika od zbędnej pracy. Wynik, jak wspomniałem, może być znacznie większy od górnej granicy 10000 obr/min, które jesteśmy w stanie wystawić na diody. Jest to bardzo cenna zaleta! Oczywiście, tym dokładniejsze będą wyniki, im wyższą wskażemy częstotliwość statycznego obrazu (mała wartość *a* – dlatego radziłem, by mierzyć, zaczynając od dużych częstotliwości stroboskopu).

Aby wykonać taki zautomatyzowany pomiar, szukamy pierwszego obrazu opisaną wcześniej metodą, po czym naciskamy przycisk *Przechwyć*. W górnym prawym rogu pojawią się dwie ikonki: prostokąta oraz trójkątnej strzałki (**fotografia 4a**). Prostokąt po prawej oznacza, że w pamięci urządzenia jest już częstotliwość pierwszego obrazu. Strzałka natomiast oznacza, że ta częstotliwość jest właśnie ustawiona na diodach i musimy szukać innego obrazu (żeby było jasne dla użytkownika: nie możemy zapamiętać dwa razy tego samego). Następnie szukamy drugiego obrazu, oczywiście możemy iść w dowolnym kierunku (większe i mniejsze częstotliwości). Zauważmy, że gdy ustawimy na diodach inną częstotliwość niż zapamiętana, natychmiast zniknie trójkątna strzałka obok prostokąta. Jeśli mamy w pamięci już zapamiętaną jakąś częstotliwość (obecność prostokąta w rogu), możemy w dowolnym momencie powrócić do niej, czyli ustawić ją na diodach. Do tego służy przycisk *Pokaż* – przydaje się, kiedy uznamy, że „zgubiliśmy” się w szukaniu najbliższego sąsiedniego obrazu.

Kiedy już znajdziemy drugi obraz, znów naciskamy przycisk *Przechwyć*. W rogu pojawi się druga prostokątna ikonka, a na wyświetlaczu ukaże się wynik – poszukiwana prędkość obrotowa (**fotografia 4b i c**). W tym momencie diody nadal błyskają z częstotliwością równą drugiej wskazanej, a my możemy wyjść z trybu wyświetlania wyniku, naciskając *Przybliż do 500*, *Czyść* lub *Przechwyć*, wyczyścimy wówczas pamięć obrazów, a ustawiona wcześniej częstotliwość i krok się nie zmienia.



Fot. 4



Fot. 5

Ale jak wspomniałem, mamy możliwość zweryfikowania poprawności wykonania pomiaru – jeśli w trybie wyświetlania wyniku nacisniemy przycisk *Pokaż* lub gałkę impulsatora, wynik ten zostanie ustawiony na diodach, rzecz jasna – po zaokrągleniu do wielokrotności kroku minimalnego. W bezpośredniej postaci zostanie on przeniesiony na diody w sytuacji z fotografii 4b – zauważmy, że w dolnym prawym rogu wyświetlacza LCD znajduje się mała trójkątna strzałka, co

oznacza, że wynik mieści się w zakresie do 10000 błysków/minutę i można (po zaokrągleniu) ustawić go na diodach. A co, gdy obliczona wartość jest za duża, jak ma to miejsce na fotografii 4c? Wtedy urządzenie informuje o tym innym wyglądem strzałki w rogu, a po naciśnięciu ośki impulsatora lub microswitcha *Pokaż* zostanie on przeskalowany, czyli podzielony przez najmniejszą liczbę naturalną, aby zmieścił się w zakresie 10000 obr/min. W obu tych przypadkach, wraz

R E K L A M A

z wynikiem na diodach, automatycznie ustawiony zostanie krok 2x. Kręcąc impulsatorem, podwajamy lub dzielimy przez 2 bieżącą wartość. Obracamy ośkę impulsatora w stronę mniejszych częstotliwości i sprawdzamy, czy i dla nich uzyskujemy statyczne obrazy. Jeśli tak, pomiar jest prawidłowy. W ten sposób możemy łatwo zweryfikować jego wiarygodność. W praktyce działa to zaskakująco dobrze, nawet dla wspomnianego wiatraka (22 tys. obr./min), wszystkie pod wielokrotnie obrazy bardzo powoli się obracały, co dobrze świadczy o obrotomierzu. Efekty trudno uchwycić na zdjęciach, na **fotografii 5** można zobaczyć moje najlepsze ujęcia – wyraźny obraz rotora wirującego wentylatora oraz drgająca struna gitary, której oświetlony odcinek zdaje się stać w miejscu. Trzeba to zobaczyć na własne oczy, a bardzo łatwo je oszukać...

Oprogramowanie

Na koniec chciałbym omówić nieco program na mikrokontroler. Nie mam zamiaru przedstawiać masy listingów, lecz opisać metodę wytwarzania częstotliwości z użyciem 8-bitowego procesorka, ponieważ może mieć wartość edukacyjną dla niektórych Czytelników. Z pewnością okaże się też pomocna tym, którzy zechcą zmodyfikować mój program bądź też napisać własny od zera. Jak wspomniałem, urządzenie daje w tym zakresie bardzo duże możliwości, ponieważ nie ma tutaj znaczących sprzętowych ograniczeń. Program napisałem w języku Bascom AVR, zajmuje 77% pamięci FLASH układu ATmega8A, kod jest bogaty w komentarze.

Do wytwarzania częstotliwości wykorzystywany jest licznik Timer0. Czego od niego chcemy: ustaliliśmy, że najwyższa wytwarzana częstotliwość to 10000 błysków/min, czyli 166,6Hz (okres 6ms). Przy tych najwyższych częstotliwościach chcemy mieć krok regulacji 50 obr./min – taki wybrałem jako optymalny, aby nie był za duży (precyzja), ani za mały (trudność i błąd generowania). Jeżeli zmniejszymy częstotliwość z maksymalnej na mniejszą o jeden mały krok, czyli z 10000 do 9950 błysków/min, to częstotliwość wyniesie 165,83Hz (okres 6,03ms). Różnica

okresów tych sygnałów (6ms, 6,03ms) wynosi około 30μs, a zauważmy, że im niżej będziemy schodzić (9900, 9850...), tym różnice między okresami kolejnych częstotliwości będą wzrastać (ale bardzo powoli). Zatem przejście 10000<->9950 bł/min to najgorszy (najtrudniejszy) przypadek i od jego rozpatrzenia musimy zacząć. Mikrokontroler może generować częstotliwość na podstawie swojego zegara, odpowiednio podzielonego (preskalowanego) przez timer i ewentualnie jakieś dodatkowe funkcje itp., ale musimy pamiętać, że nie możemy naliczyć ułamka taktu – bierzemy tylko wartość całkowitą.

Nie jest problemem zbudowanie skokowo przestrajanego generatora sygnału, nie jesteśmy jednak w stanie zrealizować w takim procesorze generatora prostokąta o w pełni liniowo przestrajanej częstotliwości (a to by nam się teraz przydało). Musimy zatem pogodzić się z pewną nieliniowością i odchyłkami wytwarzanej częstotliwości, jednak jest to pod kontrolą i błąd nie będzie duży. Wracając zatem – musimy tak skonfigurować źródło przerw (timer), by kwant jego przestrajania wynosił ok. 30μs.

U mnie Timer0 działa z preskalerem 8, przepełnienie nastąpi, gdy doliczy on do 38. Przy obecnym taktowaniu 12MHz, przerwanie od Timer0 nastąpi co 25,3μs. Przypomnę, że nie możemy też całej mocy obliczeniowej mikrokontrolera skierować na generację – mamy jeszcze wyświetlacz, przyciski, impulsator... Tak czy inaczej da się to wszystko ładnie ze sobą pogodzić. Urządzenie samo oblicza, ile przerw od timera trzeba zliczyć, aby odtworzyć zadaną częstotliwość z jak najmniejszym błędem. Jednocześnie obliczany jest czas świecenia diod na tle całego okresu – wypełnienie wynosi 1/40, błysk jest więc bardzo krótki (tego chcemy – mamy ostry obraz nawet przy szybkich wirnikach!).

Jak wspomniałem, im niżej schodzimy z częstotliwością błysków, tym różnica okresów między sąsiednimi częstotliwościami wzrasta. W pewnym momencie będzie ona już cał-

kiem spora, wtedy grzechem byłoby nie zmniejszyć kroku! Z kilku matematycznych równań (których tutaj oszczędzę) można wyznaczyć, dla jakich częstotliwości błysków można zmniejszyć krok do założonej, nowej wartości, aby różnica okresów zbliżyła się do tych 25μs (z pewnym inżynierskim zapasem). W ten sposób krok minimalny jest zależny od częstotliwości, i im niżej zejdziemy z jej wartością, tym precyzyjniej możemy ją regulować.

Na koniec mogę jeszcze podzielić się swoimi przemyśleniami co do częstości przerw od Timer0. Wiemy, że im jest większa, tym mniejszy błąd generowanej częstotliwości. Oczywiście nie możemy przesadzić – w pewnym momencie mikrokontroler nie nadąży z wykonywaniem pozostałych zadań (które, w momencie wystąpienia przerwania, są pauzowane, bo mają niższy priorytet) i urządzenie zacznie powoli i z opóźnieniem reagować na nasze polecenia („lagować”). Doświadczyłem tego, kiedy zaczynałem od taktowania 8MHz. Dlatego też, choć miałem taki pomysł, zrezygnowałem z przestrajania licznika Timer0, a ściślej – ładowania mu co przerwanie obliczonej automatycznie liczby do rejestru. Postanowiłem, że będzie ona stała (czyli wspomniane 38), a obliczana będzie tylko liczba przerw. W ten sposób urządzenie zawsze pracuje tak samo sprawnie (płynnie). Zachęcam jednak Czytelników, którzy zechcieliby trochę przy kodzie posiedzieć i zrealizować takie dokładniejsze korygowanie generatora – zmniejszyłby się wtedy błąd. Można też spróbować wykorzystać szybszy procesor, ale to wymaga dużych zmian. Nie jest to konieczne, ponieważ obrotomierz i w takiej wersji jest urządzeniem dość dokładnym. Ale przecież zawsze można go udoskonalić – najwięcej zyskamy na pomiarach dużych szybkości obrotowych, a zwłaszcza tych wysoko ponad zakresem częstotliwości, które możemy zrealizować na diodach.



Michał Pędzimąż
mpedzimaz@gmail.com

R E K L A M A