kit 2740 AVT

Intrygujący Tęczowy kryształ

Hm... „Elektronika dla Wszystkich". Tytuł troszkę zobowiązuje. Duży nakład sprawia, że wśród Czytelników znajdują się zapewne ludzie o bardzo zróżnicowanych, czasem zakręconych, pozytywnie zakręconych, zainteresowaniach. Układ, który mam zamiar przedstawić, mam nadzieję, że zainteresuje najbardziej właśnie „pozytywnie zakręcone osoby". Ale nie tylko. Uzyskany efekt jest na tyle ciekawy, że powinien zaspokoić także gusta Czytelników lubiących oryginalne a przy tym estetyczne ozdoby.

Aby zakończyć już budowanie napięcia, zacznę pisać o samym układzie. Ciekawostką jest to, że urządzenie powstało na zamówienie firmy zajmującej się ezoteryką. Aktualnie jednak, ponieważ zachowanie tajemnicy konstrukcyjnej okazuje się nieistotne, mam możliwość przedstawienia tego ciekawego rozwiązania.

Układ steruje trójkolorową diodą w taki sposób, aby uzyskać powolne i płynne przechodzenie pomiędzy kolorami. W artykule opiszę także sposób ręcznego wykonania eleganckiej i pasującej do układu obudowy. Chociaż całość została pomyślana do podświetlenia kryształu, nie wątpię, że Czytelnicy znajdą także inne możliwości wykorzystania.

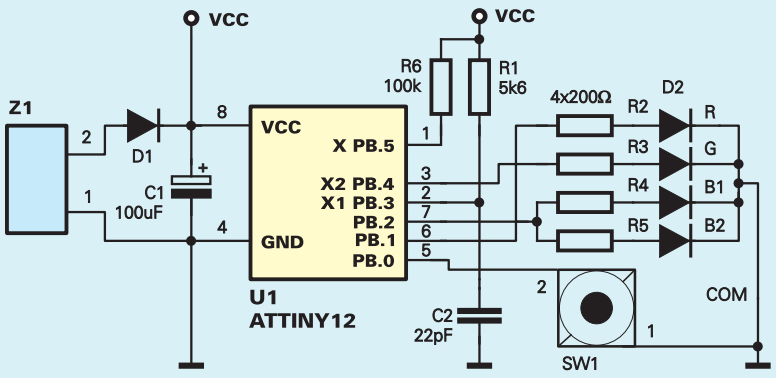
Poza płynnymi przejściami między kolejnymi kolorami istnieje także możliwość wybrania tylko jednego koloru, który w takim przypadku zaczyna pulsować. Tutaj otwiera się właśnie możliwość wykorzystania układu do wsparcia terapii kolorami. Zagadnienie jest dość obszerne i zupełnie niezwiązane z tematyką tego pisma. Myślę, że osoby zainteresowane tematem albo już wiedzą o co chodzi, albo też nie będą miały problemów z dotarciem do odpowiednich materiałów.

Opis układu

Jednym z założeń zlecenia było, aby układ elektroniczny był możliwie tani. W pierwszej chwili zacząłem analizować różne możliwości wykorzystania przestrajanych generatorów zbudowanych na bramkach, wzmacniaczach operacyjnych... nawet tranzystorach. Szybko jednak okazało się, że nie tędy droga. Urządzenie stawało się coraz większe, coraz więcej części, co prawda tanich, ale tworzących w sumie coraz większą cenę. Coraz większa płytka drukowana, co znów ma odbicie w cenie. A nadal jeszcze nie było możliwości wybrania koloru...

I tak też w pewnej chwili stwierdziłem, że nie tędy droga... Wyjściem będzie zapewne jakiś ośmiokońcówkowy procesorek. W miarę tani. Taki, który znam. Droga wiodła przez rodzinę AVR firmy ATMEL aż do zaskakująco taniego ATtiny12 (kupiłem go za 6zł+VAT).

Rys. 1



Od tej chwili projekt ruszył szybko do przodu. Schemat pokazany na **rysunku 1** powstał na przysłowiowym kolanie. Prototyp wykonany na płytce uniwersalnej był gotowy już po 30 minutach od rozpoczęcia pracy.

W samym układzie elektronicznym nie znajdziemy żadnych rewelacji. Dioda D1 zabezpiecza urządzenie przed odwrotną polaryzacją zasilania. Kondensator C1 ma za zadanie filtrowanie napięcia zasilającego. Ponieważ częstotliwość taktowania układu z wewnętrznego oscylatora okazała się stanowczo zbyt mała (migotanie diody), konieczne było wykorzystanie zewnętrznego oscylatora. Ze względy na oszczędności tak końcówkowe, jak i finansowe, skorzystałem z rezonatora RC. Przy elementach R1 i C2, jak na schemacie, częstotliwość pracy wyniesie około 4MHz. Jej stabilność czy dokładność nie ma znaczenia.

Podpięcie diody trójkolorowej D2 i przycisku sterującego SW1 nie powinno budzić żadnych wątpliwości. Zastanowić może zastosowanie rezystora R6. Podłączenie wejścia zerowania do stanu wysokiego okazało się konieczne. Jednakże aby nie zablokować sobie możliwości programowania w systemie (wykorzystywałem zestaw AVT871), podłączenie to nie mogło być „sztywne". Rezystor R6 podciąga wejście zerowania do stanu wysokiego, nie blokując jednocześnie możliwości wprowadzenia procesora w tryb programowania.

Do tej chwili wszystko było proste. Decyzje podejmowane natychmiastowo. Problem pojawił się później.

Niemożliwe okazało się wykorzystanie kompilatora BASCOM czy bardzo przeze mnie lubianego AVR-GCC. Procesor ten ma na tyle ograniczoną strukturę, że żaden język wysokiego poziomu nie ma ochoty generować dla niego kodu. Asembler na szczęście nie ma takich wymagań. Większe wymagania stawia natomiast programiście.

Program powstawał więc w wielkich trudach i bólach. Wykorzystałem udostępnione przez firmę ATMEL procedury mnożenia i dzielenia. Te drugie zostały zmodyfikowane, ponieważ nie mogłem znaleźć procedury dzielenia liczby 16-bitowej przez 8-bitową. A wykorzystanie w tym miejscu dzielenia typu 16 / 16 to zbyt wielkie marnowanie niewielkich przecież zasobów.

Tak kod źródłowy, jak i wynikowy udostępniam na stronie internetowej *Elektroniki dla Wszystkich* w dziale FTP. Tutaj postaram się wyjaśnić jego działanie.

Z naszego punktu widzenia program można podzielić na dwa ważne wątki, pracujące całkowicie niezależnie.

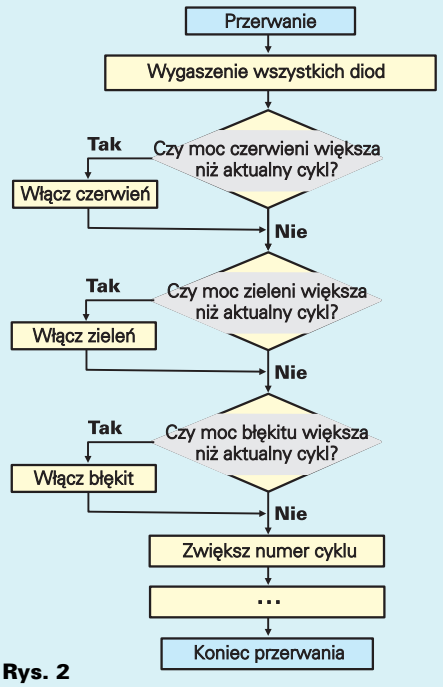
Pierwszy z nich wywoływany jest przez przerwanie z zegara/licznika 0. Odpowiada za takie wysterowanie diody świecącej, aby sprawić wrażenie jej świecenia w odpowiednim kolorze. Wykorzystuje on do tego celu 4 wewnętrzne rejestry procesora. Rejestr nazwany **cykl** jest zwiększany po każdym przerwaniu. Pełen cykl pracy obejmuje 256 wywołań podprogramu. Rejestry oznaczone **red, green, blue** zawierają aktualnie ustawioną moc świecenia poszczególnych składowych. Idea pracy jest pokazana na **rysunku 2.** W wykropkowanej części znajduje się procedura ułatwiająca generowanie opóźnień.

Timer 0 został tak ustawiony, aby generować przerwanie ponad 15 tysięcy razy na sekundę. Okazało się to konieczne ze względu na sposób sterowania. Przy 256 wywołaniach na cykl daje to częstotliwość migania diody troszkę ponad 60Hz. W związku z tym dopiero tak wysokie ustawienia zapewniają brak migania diody w subiektywnym odczuciu.

Drugi ważny wątek odbywa się w pętli głównej. Tutaj obsługiwane są płynne przejścia kolorów, szumnie nazwane w programie „animacją". Wywoływane są także procedury obsługujące naciśnięcie przycisku.

Dla mojej własnej wygody animacje zostały utworzone w postaci tabel zapisanych w pamięci programu. Do wygodnej obsługi wykorzystałem dwa rodzaje tabel. Pierwsza z nich definiuje przejścia między kolorami, druga zawiera adresy do początków poszczególnych programów. Każda z tabel kończy się znakiem 0. Informuje on o końcu danych.

Dwie tabelki zostały przedstawione na **listingu 1.** Pierwsza zawiera definicje animacji z przejściem między poszczególnymi kolorami. Druga to tablica wszystkich możliwych programów. Druga tabela nie wymaga raczej specjalnego tłumaczenia. Niektórych może zaskoczyć tylko mnożenie odpowiednich adresów przez 2. Otóż w procesorach rodziny AVR, jak wiadomo, pamięć jest zorganizowana w 16-bitowe słowa. W związku z tym adres ukryty pod odpowiednią etykietą jest właśnie numerem słowa od początku



Rys. 2

Listing 1

```
BASE_KOLOR:
; Format - Długość przejścia (cykle), RGB
;    Czas,   R,    G,    B
.DB 254, 255,    0,    0 ;Czerwień
.DB 254, 255, 128,    0 ;Pomarańcz
.DB 254, 255, 255,    0 ;Żółć
.DB 254,    0, 255,    0 ;Zieleń
.DB 254,    0, 255,   63 ;Przedłużenie zieleni
.DB 254,    0, 255, 255 ;Niebieski
.DB 254,    0,    0, 255 ;Indygo
.DB 254, 128,    0, 255 ;Fiolet
.DB 254, 255,    0, 255 ;Lawenda
.DB 254, 255, 255, 255 ;Biel
.DB 0
...

PROGRAMS_TABLE:
.DW 2*BASE_KOLOR, 2*CZERWONY_KOLOR, 2*POMARANCZOWY_KOLOR,
2*ZOLTY_KOLOR
.DW 2*ZIELONY_KOLOR, 2*NIEBIESKI_KOLOR, 2*INDYGO_KOLOR,
2*FIOLETOWY_KOLOR
.DW 2*LAWENDOWY_KOLOR, 2*BIALY_KOLOR, 0
```

pamięci programu. O ile przy rozkazach skoku nie ma problemu – wszystkie one wykorzystują numer słowa, o tyle przy rozkazie pobrania danej z pamięci programu wykorzystywany jest numer bajtu. Przemnożenie etykiety przez 2 rozwiązuje wszelkie problemy.

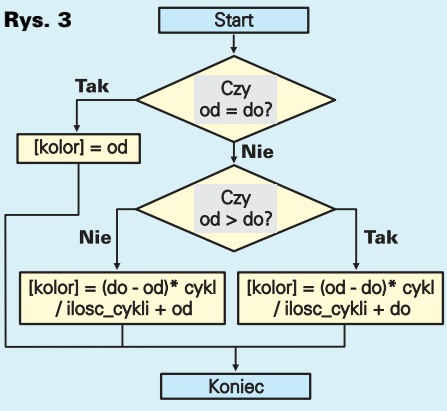
Osobnego omówienia wymaga pierwsza tabelka. Zdefiniowano w niej kolejno: czas przejścia animacji, początkowy kolor czerwony, zielony, niebieski. Największa możliwa wartość ustawionego czasu wynosi 254 cykle. Wybranie 255 spowoduje zatrzymanie się programu. Każdy cykl animacji to około 1/60 sekundy.

Troszkę o interpretacji przez program tabeli animacji. Do płynnych przejść wykorzystywanych jest 8 rejestrów (poza rejestrami służącymi do obsługi samych tabel). Po dwa na każdy kolor, oznaczające, od jakiej wartości zmierzać i na jakiej zakończyć. Zostały im nadane nazwy **red_from, red_to, green_from(...).** Poza tym znaczenie mają rejestry oznaczone jako **klatka i ilosc_klatek.** Po każdym cyklu animacji **klatka** jest zwiększana o 1. Gdy osiągnie wartość większą od **ilosc_klatek**, program wczytuje kolejną linię.

Na początek do rejestrów **[kolor]_from** przepisywane są dane z pierwszej linii w tabeli. Do rejestrów **[kolor]_to** przepisywane są dane z linii następnej. W każdym cyklu aktualna wartość koloru jest obliczana według kodu blokowego na **rysunku 3.** Cała procedura została zdeklarowana jako makro i jest wywoływana trzykrotnie w każdym cyklu – raz na każdą składową koloru.

Wspomnę jeszcze tylko o przeznaczeniu i **działaniu przycisku SW1.** Po włączeniu układ rozpoczyna generowanie pierwszej animacji. Zobaczymy płynne przejścia między kolorami. Każde kliknięcie przycisku powoduje przejście do kolejnej animacji zawartej w tabeli. Są to w praktyce poszczególne kolory które w tym przypadku zaczynają wolno pulsować. Przytrzymanie SW1 spowoduje wyłączenie urządzenia. Dioda jest gaszona, aktywowane jest przerwanie na zmianę stanu wyprowadzeń, procesor przechodzi w stan uśpienia. Dzięki aktywnemu przerwaniu możliwe jest wybudzenie procesora przez kolejne naciśnięcie przycisku.

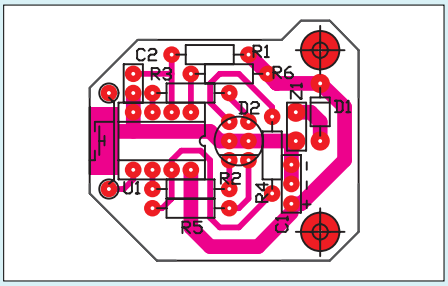
Rys. 3



Omówiłem najciekawsze, wydaje mi się, elementy programu. Oznacza to w praktyce części, które zajęły mi najwięcej pracy. Dla zainteresowanych działaniem całości jak zwykle polecam analizę udostępnionego kodu źródłowego. Pomocne będą w tym przypadku umieszczone tam komentarze.

Montaż i uruchomienie

Zmontowanie układu nie powinno przysporzyć żadnych trudności. Schemat montażowy pokazuje **rysunek 4.** Elementy montujemy od najmniejszego po najwyższy. Pod układ U1 można zastosować podstawkę.



Rys. 4

Jedynym zaskoczeniem może być sposób zamocowania przycisku SW1. W tym miejscu umieszczamy zwykły µswitch, przy czym umieszczamy go poziomo. Lutujemy dwa jego wyprowadzenia do przeznaczonych do tego pól miedzi. Powinny to być wyprowadzenia, które są wewnętrznie połączone. Następnie kawałkiem srebrzanki łączymy pozostałe dwa wyprowadzenia z punktami lutowniczymi znajdującymi się po bokach. Tak nietypowy montaż ma na celu wygodne umieszczenie całości w dość ciekawej obudowie.

Prawidłowo zmontowany układ zasilony napięciem rzędu 6V startuje od razu i nie wymaga uruchamiania. Pobór prądu nie przekracza 40mA.

Obudowa

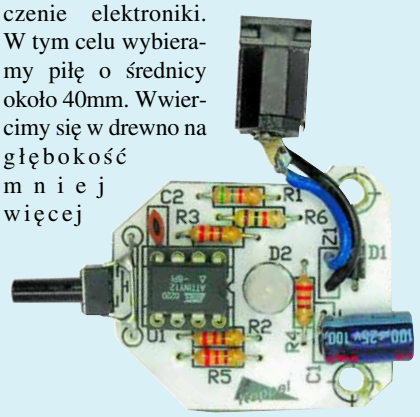
Układ bardzo zyskuje na walorach estetycznych po umieszczeniu w eleganckiej obudowie. Ponieważ nie udało mi się odnaleźć nic wystarczająco ładnego pośród obudów dostępnych na rynku, podjąłem się samodzielnego jej wykonania. Efekt tego działania można zobaczyć na zdjęciach w artykule.

Mimo tego, że jestem typowym elektronikiem i raczej stronię od prac mechanicznych, muszę stwierdzić, że ku swemu zdziwieniu, z wykonywania tej obudowy czerpałem sporo radości. Tego samego życzę Czytelnikom, którzy zechcą skorzystać z poniższego opisu.

Jako materiał do wykonania obudowy potrzebna będzie deska z litego drewna o grubości około 18mm. Może być to ewentualnie drewno klejone, często sprzedawane po atrakcyjniejszej cenie. Powinniśmy podziękować jednak za wszelkiego rodzaju płyty i sklejki.

Teraz część dla bardziej wytrzymałych psychicznie Czytelników – niezbędne narzędzia. Potrzebna nam będzie otwornica do drewna. Jest to rodzaj piły służącej do wycinania w drewnie otworów o sporych średnicach. Sprzedawana jest zwykle pod postacią przystawki do wiertarki elektrycznej. Jest to raczej urządzenie niedrogie. Poza tym przydatne będą także dłutka do drewna, papier ścierny, pilniki – iglaki.

Na początku wykonamy otwór na umieszczenie elektroniki. W tym celu wybieramy piłę o średnicy około 40mm. Wwiercimy się w drewno na głębokość mniej więcej



10mm. Warto zaznaczyć odpowiedni poziom taśmą klejącą na pile. Inaczej wielkim zaskoczeniem może okazać się, że drewno się już skończyło... a my musimy próbować jeszcze raz.

Po wykonaniu takiego otworu przychodzi pora na wykorzystanie dłutek. Za pomocą lekkiego młotka usuwamy wnętrze naciętego otworu, tak aby w środku pozostało puste miejsce, w którym umieścimy całą elektronikę.

Gdy i tę pracę zakończymy, zakładamy piłę o średnicy 60mm i wycinamy kółeczko zawierające nasz otwór. **Uwaga. Ponieważ całe wnętrze zostało usunięte, nie zadziała zawarta w otwornicy sprężyna. Ważna jest więc duża ostrożność i delikatność przyłożenia piły. W innym przypadku szarpnięcie wiertarki może być w najlepszym przypadku bolesne.**

Dalszy etap to już dopasowanie otworu za pomocą pilników i dłutek, tak aby płytka wchodziła do środka na wcisk. Pod wyprowadzenie przycisku piłujemy otwór za pomocą niewielkiego, okrągłego iglaka. Ważne jest także wypiłowanie otworu pod gniazdko zasilające. Jego kształt zależy od użytego elementu. Ja wykorzystałem element do montażu na płytce drukowanej. Jego kształt okazał się bardzo wygodny.

Gdy wszystko zostanie już dopasowane, warto jeszcze wykonać jakieś zamknięcie obudowy. Ja posłużyłem się cienkim plastikiem, który da się ciąć nożyczkami. W ostateczności można wykorzystać nawet tekturę. Wycinamy kółeczko o rozmiarach obudowy. Można zamocować je za pomocą wkrętów do drewna. Ze względu na niewielką grubość ścianki ważne jest nawiercenie otworów pod wkręty. Ich wkręcanie na siłę spowoduje pęknięcie brzegów obudowy.

Na koniec warto wszystko przeszlifować papierem ściernym i polakierować. W dobrze wykonanej obudowie wszystkie elementy wchodzą „na ścisk". Dioda świecąca powinna wpasować się idealnie w otwór wykonany przez piłę. Zbędne jest jakiekolwiek klejenie czy przykręcanie. Jeśli okaże się to jednak konieczne, można oczywiście posłużyć się klejem.

Dla zaawansowanych – możliwości zmian

Poza wszelkimi możliwościami, jakie daje samodzielne wykonanie obudowy – szczególnie esteci mogą przyozdobić ją na wiele sposobów, istnieje możliwość zmiany generowanych efektów. W tym celu należy wyedytować tabele animacji zawarte w programie. Można dodać nowy program. Należy zdefiniować wtedy nową tabelę animacji i dodać ją do tabeli programów.

Okazuje się, że układ ma problemy z uzyskaniem koloru białego. Można poeksperymentować jeszcze z doborem rezystorów ograniczających prąd. Moje doświadczenie wykazuje jednak, że eksperymenty są żmudne, a efekt niewielki. W praktycznym układzie niedostatek ten nie jest bardzo dokuczliwy.

Na zakończenie pozostaje mi jedynie życzyć dużo zabawy przy wykonywaniu układu, radości z jego obserwowania... a może i udanych medytacji...

Radosław Koppel
radoslaw.koppel@edw.com.pl

Wykaz elementów

Rezystory
R1 ..... 5,6kΩ
R2-R5 ..... 200Ω
R6 ..... 10kΩ

Kondensatory
C1 ..... 100µF
C2 ..... 22pF

Półprzewodniki
U1 ..... ATtiny12 (zaprogramowany)
D1 ..... Dioda prostownicza
D2 ..... Dioda trzykolorowa RGB

Inne
SW1 ..... µswitch 10mm
Gniazdko zasilania (patrz opis obudowy)

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2740.