kit 2881 AVT

Tajemnicza kula

– 385 diod LED,
– 25 płytek drukowanych,
– przestrzenna realizacja,
– sterowanie Attiny2313; 6 różnych efektów, możliwość zaprogramowania własnych efektów świetlnych.

Trudno bezpośrednio przedstawić zakres zastosowań prezentowanego układu. Jest on, podobnie jak większość tego typu projektów… uniwersalny. To, gdzie ten efektowny gadżet zostanie umieszczony, zależy wyłącznie od pomysłu właściciela. Przy zwykłych diodach LED (o słabej światłości) najlepszy efekt uzyskuje się w ciemności. Mimo to przy świetle dziennym też wygląda interesująco, a jeśli komuś będzie mało, zawsze może zastosować diody lepszej jakości.

Jak to działa?

Schemat sterownika pokazany jest na **rysunku 1**, a wyświetlacza na **rysunku 2.** Jak widać, sercem układu jest mikroprocesor Attiny2313, który steruje całym układem. Najważniejszą sprawą jest to, że układ nie jest matrycą diod. Początkowo nie miał być sterowany mikroprocesorem, dlatego jest on w istocie... linijką LED-ów (9 x 48 diod). Najmniejsza liczba diod, jaka w danym momencie może się zaświecić, to 48 (po dwie diody na każdej płytce wyświetlacza).

Program

Program został napisany w języku Bascom i można go ściągnąć z Elportalu.

Na początku typowo następuje deklaracja zmiennych. Następnie należy skonfigurować porty. Jako wyjścia, które sterują pracą diod, został użyty cały PortB i PortD.6, natomiast końcówki PD.0...PD.5 są skonfigurowane jako wejścia, do których należy podłączyć przyciski S1... S6. Oprócz tego, na początku programu nadajemy jeszcze wartość 2 zmiennej *Przyspieszanie*, ponieważ zmienna ta nie może mieć wartości 0.

Po konfiguracji portów, odblokowujemy przerwania i Timer0. Podział timera ustawiamy na 64, co da przerwanie co ok. 4ms.

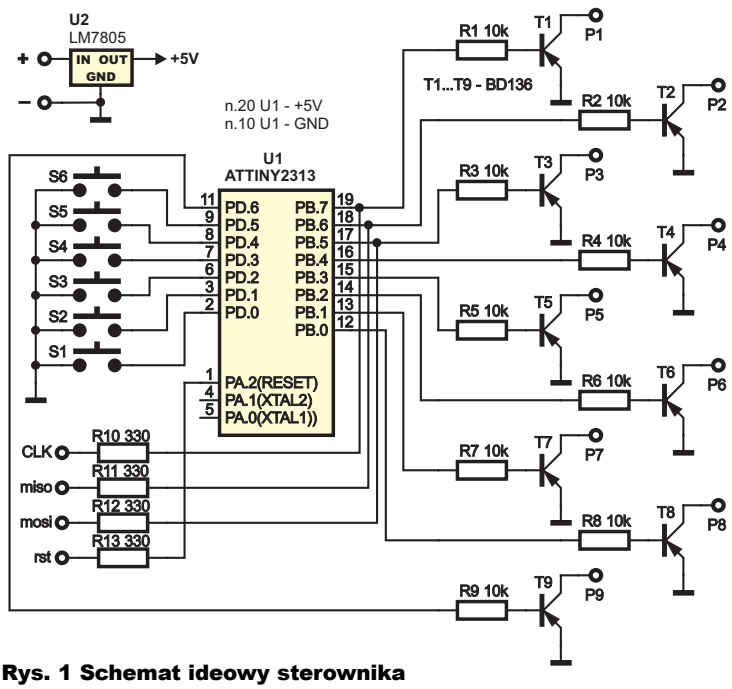
Po każdym zgłoszonym przerwaniu jest wykonywany program umieszczony pod procedurą „Co4ms". Na samym początku tej procedury program przeskakuje do procedury obsługi klawiatury. Jeśli któryś z przycisków jest naciśnięty przez czas trwania więcej niż ośmiu przerwań, to (dopiero) wtedy zostaje do zmiennej *Efekt* wpisana liczba 1...6 (w zależności od tego, który z przycisków S1... S6 został naciśnięty). Program czeka „aż tyle czasu", aby wyeliminować drgania styków przycisków. Jest to ok. 32ms, ponieważ przerwanie jest zgłaszane co ok. 4ms, a liczba powtórzeń >8 (4ms * >8 = >32ms).

Procedura obsługi klawiatury opiera się na sprawdzaniu, który przycisk jest naciśnięty (na którym wejściu procesora panuje stan niski) i powiększaniu, w zależności od przycisku, który jest naciśnięty zawartości zmiennej S1...S6. Jeśli zawartość którejś ze zmiennych będzie większa od 8, to do zmiennej *Efekt* zostanie wpisana (w zależności od tego, jaki przycisk był naciśnięty) liczba 1...6, natomiast jeśli któryś przycisk zostanie zwolniony, to zawartość zmiennej „powiązanej" z tym przyciskiem jest zerowana. Jeśli były to tylko drgania styków i zawartość zmiennej została wyzerowana, zanim osiągnęła wartość >8, to program nie zwróci na to uwagi. Jeśli natomiast przycisk był naciśnięty dłużej niż ok. 32ms (musiało to być celowe naciśnięcie), to program po zwolnieniu przycisku również wyzeruje powiązaną z nim zmienną, jednak zanim to zrobi, zmieni także zawartość zmiennej *Efekt*, więc to naciśnięcie przycisku zostanie zauważone.

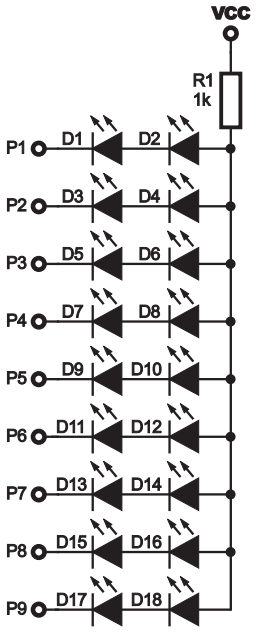
Każda sekwencja zaświecania diod została umieszczona pod odpowiednią procedurą – *Efekt1*: ... *Efekt6*:.

Procedura *Efekt1*: po naciśnięciu przycisku S1 jest wykonywana co 4ms, a to zapewniałoby zbyt szybką prędkość migania. Dlatego na początku znajduje się instrukcja, która odpowiada za zmniejszenie tej prędkości. Jeśli zawartość zmiennej C1 jest większa od 10, to zwiększamy zawartość zmiennej Cc1 i zerujemy zmienną C1, jeśli jednak zawartość zmiennej C1 nie jest większa od 10, to zwiększamy zawartość tej zmiennej o 1 co 4ms, aż do momentu, gdy tę wartość osiągnie, a wtedy zmienna ta jest zerowana i zabawa zaczyna się od początku... Dzięki temu zawartość zmiennej Cc1 jest zwiększana o 1 co ok. 44ms (>10 =11; 11*4ms=44ms), a za tym idzie fakt, że kolejny rząd jest zaświecany po 44ms (a poprzednio świecący się jest gaszony). W zależności od tego, jaką wartość przyjmie zmienna Cc1, za pomocą instrukcji Select Case wybiera się, który rząd diod można zaświecić. Wartość zmiennej rośnie w kolejności: 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3..., więc kolejność zaświecania diod najłatwiej jest zmienić zmieniając fragment umieszczony pod Case1...Case9. Jako poradę przy edytowaniu programu podaję, że PortB=&B-rząd1-rząd2-rząd3-rząd4-rząd5-rząd6-rząd7-rząd8-; PortD.6-rząd9, 0 – świeci się, 1 – nie świeci się.

Najłatwiej jest zmienić kolejność zaświecania kolejnych grup diod, zmieniając cyferki pod kolejnymi Case'ami w poleceniu Selekt Case. Tak też właśnie został utworzony *Efekt2*. Od *Efektu1* różni się on kierunkiem przewijania się punktu świetlnego, co można łatwo zauważyć, porównując Case'y z tych dwóch procedur. Tak samo powstał też *Efekt4*, tylko, że w jego wypadku zmieniono kierunek względem *Efektu3*. Natomiast sam *Efekt3* od dwóch pozostałych różni się tym, że naraz świeci się cztery razy po dwa rzędy diod i raz jeden rząd – środkowy. Efekt ten polega na tym, że punkt świetlny nie przewija się od prawej do lewej, tak jak poprzednio, tylko pojawia się najpierw na środku, a później rozpływa jednocześnie w prawą i lewą stronę. W *Efekcie4* odwrotnie.



Rys. 1 Schemat ideowy sterownika



Rys. 2 Schemat ideowy wyświetlacza

mienie powolnego zmniejszania zmiennej *Przyspieszanie* i zwiększanie prędkości migania diod. Gdy zmienna *Przyspieszanie* przyjmie wartość mniejszą od 2, to zmienna *Przysp* przyjmuje wartość 1, co w konsekwencji powoduje zwiększanie zawartości zmiennej *Przyspieszanie* i zmniejszenie prędkości migania diod. W *Efekcie5* diody zaświecają się w takiej kolejności jak w *Efekcie4*, natomiast w *Efekcie6* podobnie jak w *Efekcie1*, z tym, że jednocześnie świeci się po dwa rzędy diod.

Montaż i uruchomienie

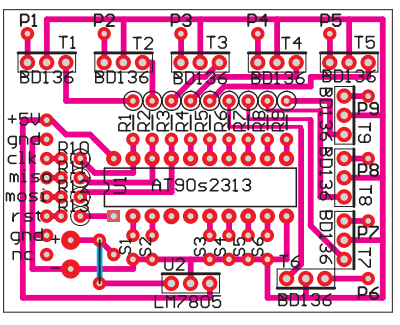
Konstrukcja składa się z 24 płytek wyświetlacza i jednej płytki sterownika. Sterownik wraz z zasilaniem jest umieszczony wewnątrz kuli. Schemat montażowy sterownika przedstawiony jest na **rysunku 3**, a wyświetlacza na **rysunku 4**. Sześć microswitchów nie jest lutowanych bezpośrednio w płytkę, tylko połączonych z nią przewodami.

Każda płytka wyświetlacza powinna składać się z 18 diod LED i jednego rezystora 1kΩ. Dlaczego tylko powinna? Jak wiemy, płytki wyświetlacza mają tworzyć kulę, a więc skoro jest ich 24, łączymy je pod kątem 15 stopni (24 *15stopni = 360stopni – koło się zamyka). Im bliżej wierzchołka kąta, tym mniejsza odległość pomiędzy ramionami, a więc diody na górze i na dole płytki wyświetlacza (diody „na górze" i „na dole" kuli są najbliżej siebie) kolidowałyby ze sobą i nie można byłoby utworzyć kuli. Z tego powodu do co drugiej płytki nie należy lutować dwóch diod na górze i dwóch na dole. A więc 12 płytek zawiera komplet 18 diod, a 12 płytek ma o cztery mniej (14).

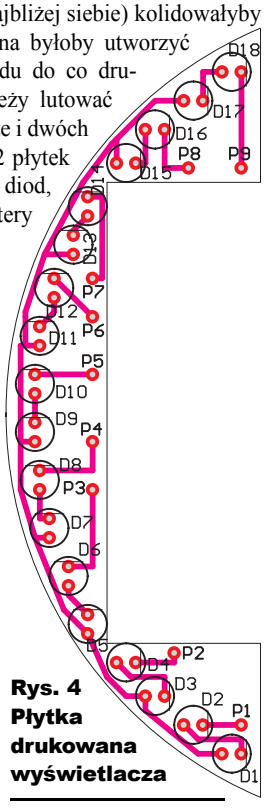
Ciąg dalszy na str. 56

Najpierw po prawej i lewej, a później przesuwa się aż do „połączenia" na środku.

Kolejna „innowacja" została wprowadzona w *Efekcie5* i *Efekcie6*. Jest to zmiana prędkości migania diod. Aby zwiększyć prędkość migania, zawartość zmiennej Cc1 jest zwiększana wtedy, gdy zawartość zmiennej C1 będzie równa wartości zmiennej *Przyspieszanie*, która z kolei przyjmuje wartości od 2...8. Gdy zmienna *Przyspieszanie* ma wartość 2, to żeby zawartość zmiennej Cc1 została zwiększona o 1, zmienna C1 potrzebuje osiągnąć tylko wartość >2. Gdy zmienna *Przyspieszanie* przyjmuje wartość 8, to zmienna Cc1 jest zwiększana o 1 dopiero po osiągnięciu wartości >8 przez zmienną C1, mija więc więcej czasu. Zmienna *Przyspieszanie* zmienia swoją wartość co drugie zwiększenie zmiennej Cc1. Daje to dosyć płynną i ładnie wyglądającą zmianę prędkości migania diod. Aby prędkość migania nie zwiększała się do maksymalnej i nagle spadała do minimalnej, zastosowałem zmienną *Przysp*. Zmienna ta przyjmuje wartość 0, gdy zmienna *Przyspieszanie* przyjmie wartość większą od 8 i powoduje uruchomienie powolnego zmniejszania zmiennej *Przyspieszanie* i zwiększanie prędkości migania diod. Gdy zmienna *Przyspieszanie* przyjmie wartość mniejszą od 2, to zmienna *Przysp* przyjmuje wartość 1, co w konsekwencji powoduje zwiększanie zawartości zmiennej *Przyspieszanie* i zmniejszenie prędkości migania diod. W *Efekcie5* diody zaświecają się w takiej kolejności jak w *Efekcie4*, natomiast w *Efekcie6* podobnie jak w *Efekcie1*, z tym, że jednocześnie świeci się po dwa rzędy diod.



Rys. 3 Płytka drukowana sterownika



Rys. 4 Płytka drukowana wyświetlacza

Ciąg dalszy ze str. 53

Dodatkowo z jednej strony kuli umieściłem diodę LED o średnicy 20mm, aby zapełnić otwór, który w tym miejscu powstał. W drugim takim otworze należy umieścić microswitche, służące do wyboru efektu.

Układ po zmontowaniu i włożeniu do podstawki zaprogramowanego procesora powinien od razu pracować. Montaż elementów w płytce sterownika jest klasyczny – od najniższych do najwyższych elementów. Rezystory lutujemy pionowo, aby uzyskać jak najmniejsze rozmiary płytki. Jeśli chodzi o płytki wyświetlacza, to aby szybko wlutować wszystkie elementy, najlepiej najpierw do każdej płytki przylutować rezystor. Warto sprawdzić, czy nie ma przerwanych ścieżek i wadliwych diod, ponieważ po połączeniu tych płytek w całość, jakakolwiek wymiana elementów będzie bardzo utrudniona. Na jednej płytce wyświetlacza znajduje się tylko jeden rezystor, mimo że większość osób zaleca inaczej, uznałem za bezsensowne stosowanie w układzie kilkuset rezystorów. Po przylutowaniu rezystorów do każdej płytki wkładamy diody i zaginamy ich nóżki, tak by trzymały się w miarę stabilnie. Gdy wszystkie są na swoich miejscach, bierzemy się do lutowania. Aby płytki utworzyły kulę, należy połączyć je ze sobą za pomocą sztywnego przewodu, pod kątem ok. <15 stopni. Najlepiej przewód taki jest wyjąć z tzw. skrętki telefonicznej i zdjąć z niego izolację.

Płytki wyświetlacza lutuje się najpierw jako dwie półkule po 12 płytek, po czym elektrycznie łączy je ze sobą, za pomocą jakichkolwiek (ok. 10cm) przewodów w izolacji. Natomiast jeśli chodzi o połączenie mechaniczne, możemy je wykonać w dowolny sposób, oczywiście tak, żeby można było bez problemu otworzyć kulę, np. wtedy gdy będzie konieczna wymiana baterii.

Do lutowania użyłem lutownicy na gorące powietrze. Przyśpieszyło to w pewnym stopniu całą pracę, bo nie musiałem dotykać grotem każdej końcówki z osobna. Większość płytek lutowałem spoiwem Pb-free i nie zauważyłem różnicy pomiędzy taką cyną a cyną z ołowiem. Różnica jest niezauważalna dla elektroników, którzy lutują swoje układy z elementów przewlekanych najczęściej lutownicami kolbowymi lub transformatorowymi.

Wykaz elementów

Sterownik

R1-R9	10kΩ
R10-R13	330Ω
T1-T9	BD136
U1	Attiny2313
U2	LM7805
S1...S6	microswitch
Gniazdo goldpin 8pin	
Przewód – taśma	
Złączka do baterii 9V	

Wyświetlacz (1 szt.)

R1	1kΩ
D1-D18	LED 5mm różne kolory

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2881.

Możliwości zmian

Na pewno najprostszą i efektowną zmianą będzie wykorzystanie diod o lepszej jasności i innych kolorach niż czerwony, żółty, zielony. Warto poszukać w Internecie ciekawych okazji lub kupić diody po cenach hurtowych.

Można eksperymentować z większymi diodami, np. wlutować jedną diodę 10mm w miejsce dwóch 5mm.

Zachęcam również do tworzenia własnych efektów świetlnych, przez modyfikowanie programu. Należy tylko pamiętać, że prezentowany układ nie jest matrycą diod. Początkowo miał być sterowany układem 4017, dlatego jest to po prostu „linijka" 9*48 diod. Najmniejsza liczba LED-ów, które mogą się jednocześnie zaświecić, to 48. Jeśli ktoś zechce, może przeprojektować układ, by każdą z diod można było zaświecić osobno – potrzebny będzie procesor z większą liczbą wyprowadzeń. Interesujące byłoby dołożenie kilku/kilkunastu elementów i zaprogramowanie procesora w ten sposób, żeby diody migały w takt muzyki itp.

Modyfikując układ (w tym również program), należy pamiętać, że tranzystory mają „ograniczone możliwości" i przy większych prądach być może będzie potrzebne zastosowanie radiatorów.

Adam Kulpiński
kulpina@onet.eu