

Fade to green

Do czego to służy?

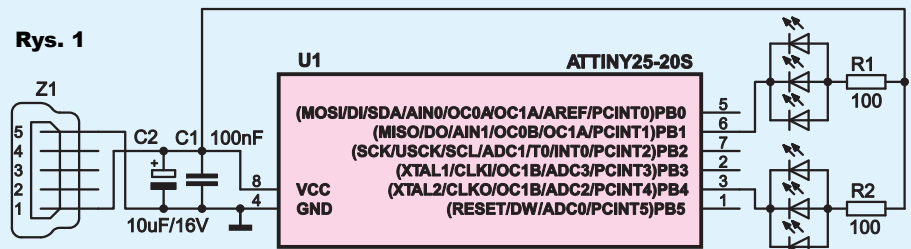
Fade to green to gadżet świetlny, w którym dwie diody LED płynnie zmieniają natężenie emitowanego światła. Jasność obu diod zmienia się zgodnie z przebiegiem sinusoidy, przy czym dla jednej z nich jest ona przesunięta o 180 stopni, dzięki czemu gdy jedna dioda się rozjaśnia, to druga się ściemnia. Najlepsze efekty daje zastosowanie urządzenia do podświetlania matowych szklanych przedmiotów, które dobrze rozpraszają światło. Układ powstał, jako rozwiązanie zadania Szkoły Konstruktorów na temat układów wykorzystywanych do pokazów i reklamowych. Stanowi też zachętę do rozpoczęcia programowania mikrokontrolerów. Jest przysłowiowym „migaczem diodą”, daje jednak ciekawy efekt. To może być pierwszy, bardzo prosty program. A gdy już umiemy zamigać diodą, co sprowadza się do cyklicznej modyfikacji zawartości rejestru mikrokontrolera, możemy zrobić o wiele więcej. Bo przecież obsługa innych peryferiów to też zmiana zawartości przypisanych im rejestrów.

I właśnie prezentowane urządzenie jest prostym przykładem wykorzystania sprzętowego generatora sygnału PWM i może być pierwszym praktycznym kontaktem z tą popularną metodą modulacji. Jest to najpopularniejsza dziś metoda regulacji jasności świecenia diod LED. Wzmacniacze audio klasy D to w sumie także modulatory PWM, tyle że o znacznej mocy. Wszelkie zasilacze impulsowe (czy to zasilacze w komputerze PC, czy małe ładowarki do telefonów komórkowych) też wykorzystują odmiany modulacji PWM. Dlatego warto zapoznać się z tą modulacją na prostym i efektywnym przykładzie z wykorzystaniem zaledwie 8 elementów i 20 linii kodu w języku C.

Jak to działa?

Na **rysunku 1** pokazany jest schemat ideowy. Układ zasilany jest z gniazda Z1 typu mini USB-B. Jest to dziś bardzo łatwo dostępne źródło napięcia 5V – układ może być dołączony do portu USB komputera lub zasilany z popularnych ładowarek stosowanych do odtwarzaczy MP3, telefonów komórkowych lub nawigacji GPS. Za sterowanie diodami LED odpowiedzialny jest mikrokontroler U1

Rys. 1



typu ATTINY25, taktowany z wewnętrznego rezonatora RC. Do obu wyjść sygnału PWM (OC1A oraz OC1B) dołączone są diody LED. Na schemacie widać sześć diod, jednak są to tak naprawdę 2 elementy mające po trzy niezależne struktury i wyprowadzenia. Program sterujący pracą urządzenia przedstawiony jest na **listingu 1**. Najpierw następuje konfiguracja pinów portu B mikrokontrolera, do których dołączone są diody, jako wyjściowych.

Następnie konfigurowany jest sprzętowy generator sygnału PWM. Dla kanału A za jego ustawienia odpowiada rejestr TCCR1, a dla kanału B – GTCCR. Generacja sygnału PWM to tylko część z licznych możliwości układu timera/licznika mikrokontrolera, dlatego aby wybrać właśnie ten tryb działania, należy ustawić bit PWM1x w przedstawionych powyżej rejestrach. Ustawienie bitów COM1x1 oraz COM1x0 powoduje, że generator sygnału PWM zostaje dołączony do wyprowadzeń mikrokontrolera. Wybiera też tryb, w którym gdy licznik doliczy do wartości wypełnienia zapisanej w rejestrze OCR1x, na wyprowadzeniu wystawiany jest stan wysoki, a dla wartości zero licznika – niski. Następnie rozpoczyna się nieskończona pętla, w której obliczane są kolejne wartości funkcji sinus i przypisywane do wartości wypełnienia sygnału PWM dla obu kanałów. Odpowiednie wartości muszą być umieszczane w rejestrach OCR1A oraz OCR1B, w postaci liczby z przedziału 0 do 255. Jednakże funkcja *sin* zwraca liczbę zmiennoprzecinkową z przedziału od -1 do 1, dlatego wynik działania tej funkcji jest sumowany z jedynką, a wynik tej operacji

```
#include <avr/io.h>
#include <util/delay.h>
#include <math.h>
double x=0,y;
int main(void)
{
  DDRB |= _BV(PB1)|_BV(PB4);
  TCCR1 = _BV(PWM1A) | _BV(COM1A1) | _BV(COM1A0) | _BV(CS10);
  GTCCR = _BV(PWM1B) | _BV(COM1B1) | _BV(COM1B0);
  while(1)
  {
    y=sin(x);
    OCR1A=(int)128*(1+y);
    OCR1B=(int)128*(1-y);
    _delay_ms(25);
    x+=0.05;
    if(x>=6.3)
      x=0;
  }
}
```

Listing 1

mnożony przez 128. Daje to liczbę z wymaganego przedziału, która musi być jeszcze zrzuconą na typ całkowity (int). Wartość dla drugiego kanału jest odejmowana od jedynki, dzięki temu zawsze wypełnienie kanału A zmienia się przeciwnie do kanału B. Szybkość zmian jasności świecenia diod można modyfikować poprzez zmianę opóźnienia pomiędzy poszczególnymi krokami (parametr funkcji *_delay_ms*) lub wartość o jaką inkrementowana jest zmienna x (argument funkcji *sin*) w każdym z kroków. By wartość zmiennej x nie rosła do bardzo dużych wartości podczas działania urządzenia jest ona zerowana po jednym okresie funkcji sinus. Przyjmuje ona argument w radianach, a wartość 6,3 to 2π w zaokrągleniu, czyli odpowiada właśnie jednemu pełnemu okresowi. Choć kod źródłowy programu jest bardzo zwięzły, to utworzony po kompilacji kod maszynowy już taki nie jest: 1278 bajtów i ponad 60% pamięci Flash wykorzystanego mikrokontrolera to całkiem sporo jak na tak proste użycie. Tak obszerny kod wynika z zastosowania obliczeń na liczbach zmiennoprzecinkowych oraz funkcji z biblioteki *math.h*. W przedstawionym zasto-

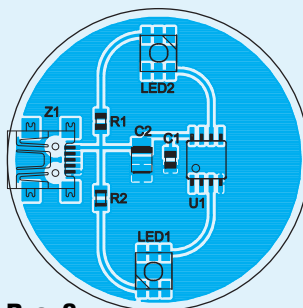
sowaniu nie stanowi to żadnego problemu, ale może być ostrzeżeniem dla początkujących programistów, że bardzo krótki kod w C może stać się bardzo rozległy, jeżeli chodzi o zajętość pamięci programu.

Montaż i uruchomienie

Schemat montażowy przedstawiony jest na rysunku 2. Wszystkie elementy układu są przeznaczone do montażu powierzchniowego SMD, także diody LED oraz kondensator C2. Elementy te można przylutować bez większych problemów. Mikrokontroler znajdujący się w kicie AVT-2984 jest już zaprogramowany. Przy samodzielnym kompletowaniu elementów **na-leży go zaprogramować przed wlotowaniem w płytkę, za pomocą specjalnego adaptera dla układów SMD, ponieważ na płycie nie przewidziano złącza programowania ISP**. Program jest przystosowany do działania z domyślnymi wartościami bitów konfiguracyjnych mikrokontrolera, więc nie ma potrzeby ich modyfikacji. Układ działa od razu po dołączeniu zasilania. Do zasilania można wykorzystać kabel USB-A – Mini USB-B i dołączyć urządzenie do portu USB komputera lub skorzystać z ładowarek do telefonów komórkowych lub innych urządzeń ze złączem Mini USB. W tym drugim przypadku warto skontrolować napięcie wyjściowe, ponieważ bez obciążenia w wielu przypadkach może ono wynosić nawet ponad 7V i nie zawsze spada pod obciążeniem do standardowej wartości 5V.

Możliwości zmian

Najprostsza zmiana to dobór diod o innym kolorze świecenia. Ponieważ diody o różnych kolorach mają różne napięcia przewodzenia, przy zmianie należy uwzględnić wartości rezystorów R1 i R2 tak, by nie przekroczyć maksymalnych prądów pracy diod i pinów zastosowanego mikrokontrolera. Rejestry sterujące pracą timerów (a zatem i generowaniem sygnałów PWM) ATmega8 różnią się nieznacznie od ATtiny25, dlatego w materiałach dodatkowych do artykułu w Elportalu umieszczona jest również alternatywna wersja programu. Do finalnej wersji projektu został wybrany ATtiny25 ze względu na miniaturyzację urządzenia oraz zmniejszenie kosztów. Wykorzystanie „większego” mikrokontrolera pozwala na modyfikacje i rozwój projektu. ATmega8 ma 3 sprzętowe kanały PWM, które można wykorzystać np. do sterowania diodą RGB i płynnej zmiany koloru jej świecenia. Można oczywiście korzystać nie tylko ze sprzętowej generacji sygnału PWM, ale też programowej, wtedy można wykorzystać tyle diod świecących, ile wyjść GPIO ma wybrany mikrokontroler. Przy tak dużej liczbie diod można się już pokusić o bardzo interesujące



Rys. 2

Wykaz elementów

R1,R2	100R
C1	100nF (0805)
C2	10 μ F/16V (SMC-B)
U1	ATtiny25-20SU
LED1,LED2	LEDP6L-GW120 (oznaczenie firmy Maritex, a nie producenta)
Z1	Mini USB

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2984.

efekty świetlne. Przy podłączeniu większej liczby diod do mikrokontrolera należy rozważyć wykorzystanie tranzystorów sterujących, bowiem maksymalny prąd wpływający do mikrokontrolera dla ATmega8 wynosi 300mA, dla ATtiny25 200mA. Przy programowym generowaniu PWM należy zwrócić uwagę na to, ile czasu zajmuje obliczanie kolejnych wartości wypełnienia. Jeżeli będzie to zbyt długi okres, sygnał może zostać zakłócony i mogą pojawić się nieprzyjemne efekty optyczne. W takim przypadku warto zrezygnować z funkcji sin i skorzystać ze stabilizowanych wartości. Po pierwszym eksperymencie z modulacją PWM i regulacją jasności świecenia diod LED można przejść np. do eksperymentów z generowaniem sygnałów dźwiękowych za pomocą tej modulacji. Życzę powodzenia!

Paweł Hoffmann

pawelhoffmann@gmail.com