Chromatyczna gwiazdka choinkowa

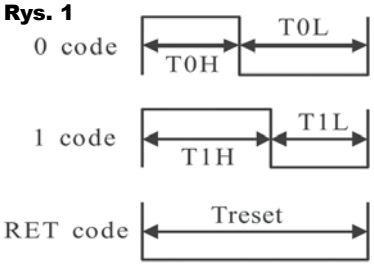
Do czego służy taki układ? Cieszy oko. Bo w zasadzie do niczego innego :). Jest to projekt typowo świąteczny. W historii EdW znalazłem podobne projekty, więc nie jest to nowa idea. W listopadzie 2001 pojawił się projekt gwiazdki złożonej z 60 dwukolorowych diod sterowanych układem logicznym. Wiele innych rozwiązań można też znaleźć, wpisując w Google->grafika hasło „led christmas star". Większość gwiazdek, podobnie jak wspomniane rozwiązanie, zrealizowane jest w postaci obrysu gwiazdy pięcio- lub sześcioramiennej. Opisywana konstrukcja jest realizacją nieco innej koncepcji, w której gwiazdę tworzą ramiona w prostej linii: sześć ramion długich i sześć krótkich. Przypomina to nieco budowę płatka śniegu. Atrakcyjność tej konstrukcji polega na wykorzystaniu diod trójkolorowych sterowanych indywidualnie, dzięki czemu można uzyskać dowolny kolor oraz dowolny wzorzec iluminacyjny.

Jak to działa?

Gwiazdka zbudowana jest z 67 diod trójkolorowych RGB typu WS2812B firmy Worldsemi. Jest to bardzo sprytny i dość zaawansowany układ, który zawiera kontroler sterujący trzema kolorowymi diodami oraz zarządzający komunikacją, a wszystko zamknięte w obudowie SMD 5050. Szczegółowy opis układu można znaleźć w Internecie (np. tu: *http://ep.com.pl/artykuly/10758--Pomyslowe_diody_LED_RGB.html*), więc wymienię tu tylko kilka najważniejszych informacji. Układ WS2812B ma cztery wyprowadzenia: zasilanie, masa, wejście danych oraz wyjście danych. Na wejście danych podaje się 24 bity informacji o kolorze świecenia, po 8 bitów na każdy z kanałów. Bity te przesyła się w konfiguracji **GRB** (zielony, czerwony, niebieski), startując od najbardziej znaczącego bitu. Wyprowadzenie z wyjściem danych umożliwia przesyłanie 24 bitów do kolejnej diody, którą podłącza się szeregowo. W ten sposób można sterować szeregiem diod z użyciem zaledwie jednej linii sterującej. Na **rysunku 1** przedstawiono podstawowe informacje dotyczące sposobu łączenia diod oraz metody komunikacji. Wspomniane 24 bity informacji przesyła się w dość specyficznym protokole komunikacyjnym. Bity są tu zakodowane w wypełnieniu przebiegu prostokątnego o okresie 1,25µs. Bit 0 zdefiniowany jest przez czas trwania stanu wysokiego równy 0,4µs, natomiast dla bitu 1 wynosi on 0,8µs.

Specyfikacja diody podaje, że czasy te mają szeroką tolerancję ±0,15µs, co pozwala na przygotowanie dość prostego sposobu sterowania z użyciem peryferii typowych mikroprocesorów.

Rys. 1



Data transfer time(TH+TL=1.25µs±600ns)

T0H	0 code, high voltage time	0.4us	±150ns
T1H	1 code, high voltage time	0.8us	±150ns
T0L	0 code, low voltage time	0.85us	±150ns
T1L	1 code, low voltage time	0.45us	±150ns
RES	low voltage time	Above 50µs	

Cascade method:

W opisywanej konstrukcji sterowanie diodami zrealizowane zostało w oparciu o popularny 32-bitowy mikroprocesor STM32F103RB. Jest to wersja mikroprocesora w obudowie LQFP64. Wybór na ten model padł z prozaicznego powodu: posiadania takowego w zasobach domowych. Do tego zastosowania z powodzeniem można było wykorzystać również wersję 48-wyprowadzeniową, choć z ekonomicznego punktu widzenia różnica jest niewielka (rzędu 2 złote). Z kolei wybór procesora z rodziny STM32 dokonany został po przeglądzie zasobów internetowych w zakresie kodów sterujących. Znaleziono sporo rozwiązań programowych w oparciu o tę rodzinę mikroprocesorów, stąd zastosowanie go w opisywanej konstrukcji dawało gwarancję powodzenia realizacji projektu.

Schemat ideowy sterownika przedstawiony jest na **rysunku 2**. **Rysunek 3** pokazuje schemat wyświetlacza. Gwiazdka zbudowana jest z dwóch płytek PCB: górnej i dolnej. Górna płytka (wyświetlacz) zawiera diody z kondensatorami odsprzęgającymi zasilanie diod, zgodnie z zaleceniami noty katalogowej. Dolna płytka zawiera procesor STM32 z regulatorami zasilania. Obie płytki połączone są przez trzy dwurzędowe złącza typu *goldpin*. Tylko jedno z tych złączy WSO doprowadza sygnały komunikacyjne do płytki z diodami. Pozostałe dwa złącza spełniają jedynie funkcję stabilnego połączenia obu płytek i nie przenoszą żadnych sygnałów.

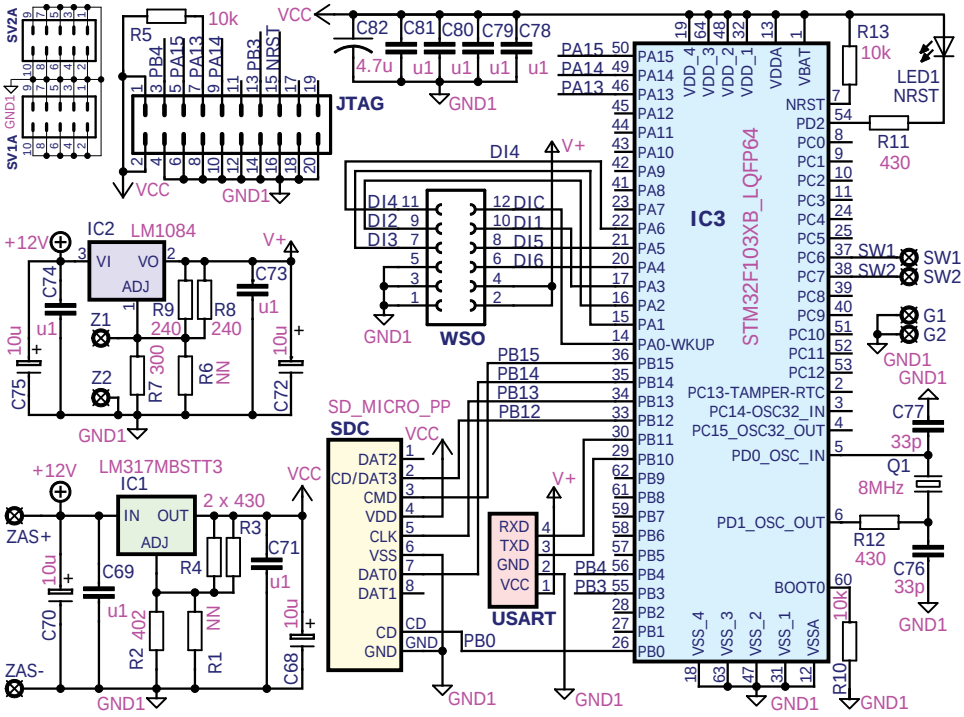
Diody sterowane są siedmioma liniami komunikacyjnymi (kanałami) obsługiwanymi przez najniższe bity portu PA mikroprocesora, przy czym PA0 (kanał 0) steruje diodą centralną, natomiast PA1–PA6 (kanały 1–6) sterują 66 diodami w grupach po 11 diod. 8 diod przypada na

długie ramię gwiazdy, a pozostałe 3 diody na krótkie ramię.

Z prostych obliczeń można określić maksymalny prąd pobierany przez wszystkie diody, który wynosi 4A (67 diod * 3*20 mA) przy najgorszym założeniu, że wszystkie diody będą wysterowane na kolor biały o największej jasności. Prąd pojedynczej diody w strukturze WS2812B został przyjęty *ad hoc* jako najbardziej prawdopodobny, gdyż specyfikacja nie podaje poboru prądu. Ze względu na tak duży prąd w projekcie zastosowano stabilizator LM1084 będący wysokoprądowym (5A) odpowiednikiem popularnego LM317.

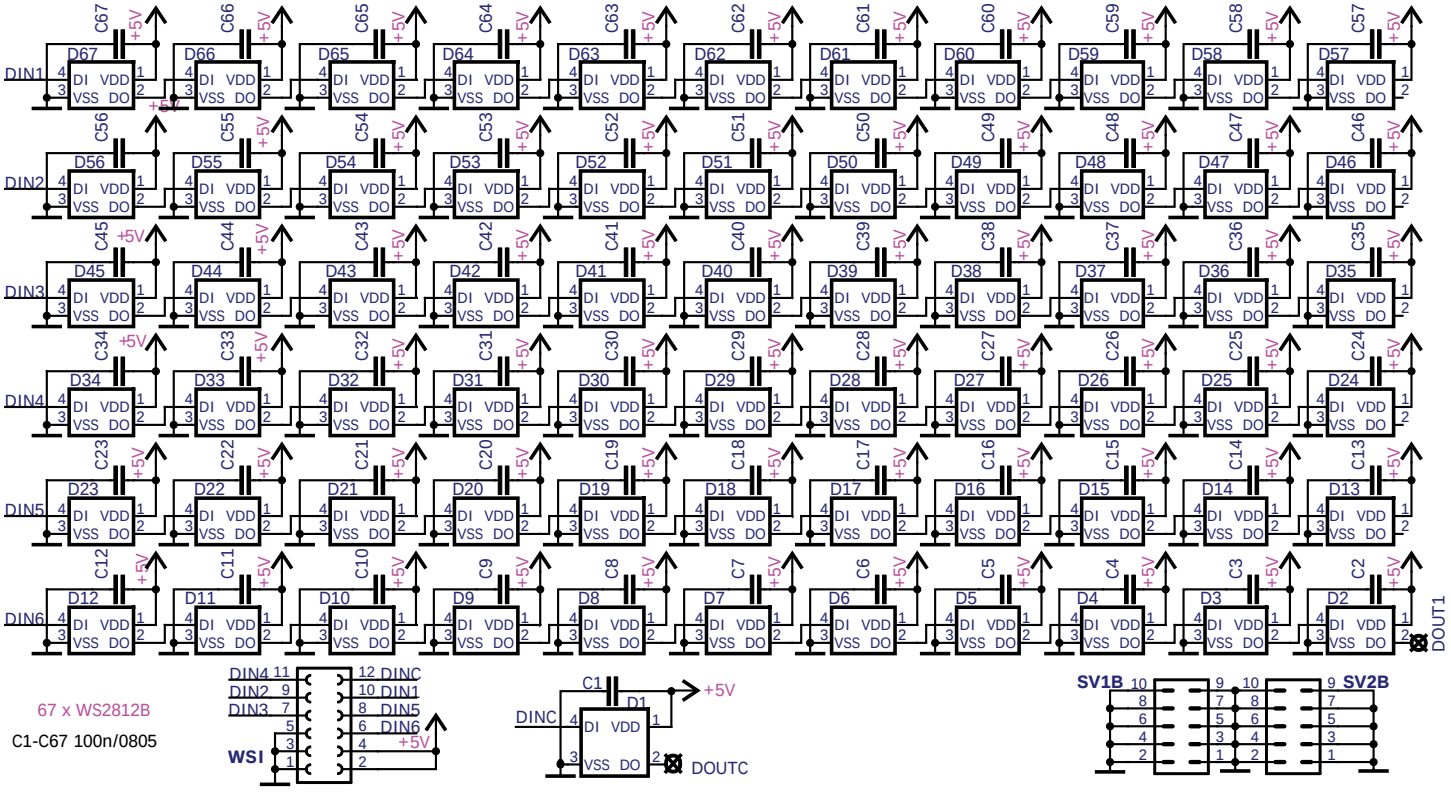
W następnej kolejności określone zostały napięcia zasilające diod oraz mikroprocesora. Najpierw należy zauważyć, że akceptowalny poziom napięcia stanu wysokiego na linii danych wejściowych DI diody wynosi 0,7*Vcc, gdzie napięcie zasilania Vcc może być w granicach 3,5–5,3 V. Przy zasilaniu diod napięciem 5V minimalne napięcie poziomu wysokiego na linii DI wyniesie więc 3,5V. Z kolei procesor STM32 może być zasilany maksymalnie do 4V. Minimalne napięcie wyjściowe na wyprowadzeniach portu PA jest trudne do określenia na podstawie specyfikacji mikroprocesora (tabela 35 w Doc ID 13587 Rev 10), ale można przyjąć, że jest ono nie mniejsze niż VDD – 0,4V. Aby napięcia na wyjściach portu PA były zgodne z akceptowalnymi napięciami wejściowymi diody WS2812B, przyjęto napięcie zasilania mikroprocesora na poziomie 3,6V, a stąd wynika minimalne napięcie dostarczane przez mikroprocesor na poziomie 3,2V. Przyjmując pewien margines tolerancji, dla diody WS2812B przyjęto więc minimalne napięcie wejściowe na linii DI na poziomie 3,1V. Stąd maksymalne napięcie zasilające diody powinno być na poziomie 3,1V/0,7, czyli nie większe niż 4,4V. Znając pożądane napięcie wyjściowe z układu LM1084 oraz LM317, dobrano odpowiednie wartości rezystorów konfigurujących.

Pozostała jeszcze kwestia ciepła wydzielanego przez stabilizator LM1084. Łatwo wyliczyć, że moc wydzielana na stabilizatorze przy maksymalnym prądzie 4A oraz przy zasilaniu stabilizatora



Rys. 2

Rys. 3



napięciem 12V wyniesie P = (12–4,4) V * 4A = 30W! Nawet przy ustawieniu minimalnego napięcia na zasilaczu impulsowym na poziomie 10,5V, wydzielana moc jest rzędu 24W. Aby odprowadzić ze stabilizatora wygenerowane ciepło przy tej mocy, należałoby zastosować dość pokaźny radiator. Tymczasem rolę radiatora odgrywa fragment płytki PCB z miedzią po obu stronach połączoną szeregiem metalizowanych otworów. Nie jest to zbyt efektywny sposób na odprowadzanie ciepła. Aby nieco poprawić parametry takiego radiatora, dolutowano trzy kawałki drutu miedzianego w celu zwiększenia powierzchni radiatora. Jednak zasadniczo problem nadmiernego ciepła rozwiązano w oprogramowaniu, zarówno poprzez ustawienie globalnej jasności wszystkich diod, jak i takie sterowanie diodami, aby nie pobierać maksymalnego prądu w zbyt długich okresach.

Program sterujący. Sterowanie diodami WS2812B zrealizowano w oparciu o kod opisany na stronie *http://www.cnblogs.com/shangdawei/p/4762035.html*. Jest to bardzo sprytna idea wykorzystująca system DMA we współpracy z licznikiem (TIM2). Układ DMA skonfigurowano tak, aby przepisywał odpowiednie dane z bufora pamięci bezpośrednio na wyjście portu PA. Sposób ten daje w efekcie 16 niezależnych kanałów umożliwiających sterowanie diodami. W każdym kanale może być podłączona dowolna liczba diod (w praktyce ograniczeniem jest ilość pamięci RAM niezbędnej do przygotowania bufora z danymi). Koncepcja generowania przebiegów sterujących diodami opiera się na spostrzeżeniu, że każdy bit informacji składa się z trzech odcinków. Pierwszy i ostatni odcinek są wspólne dla obu bitów, przy czym pierwszy odcinek jest stanem wysokim, ostatni stanem niskim. Jedynie środkowy odcinek niesie właściwą informację o bicie: stan niski w przypadku bitu równego 0 oraz stan wysoki dla bitu równego 1. Elementem odmierzającym czas trwania poszczególnych odcinków jest układ licznika TIM2. Skonfigurowano go tak, aby okres wynosił dokładnie 1250ns (800kHz), natomiast odmierzanie odcinka pierwszego oraz drugiego zostało zrealizowane z użyciem kanałów CC1 oraz CC2 (*Capture/Compare*) tego licznika. Z każdym rozpoczęciem odliczania kolejnego cyklu przez TIM2 (co 1250 ns) generowane jest żądanie DMA dla kanału drugiego (DMA_k2). Ten kanał DMA rozpoczyna budowę przebiegów od ustawienia wszystkich linii portu A na stan wysoki (pierwszy odcinek). W momencie wystąpienia zdarzenia CC1 (po ok. 350ns) uruchamiane jest żądanie DMA_k5, które przepisuje 16-bitową wartość z tablicy w pamięci RAM do portu PA (odcinek drugi). Z kolei zdarzenie CC2 (po ok. 750ns) ustawia wszystkie bity portu A w stan niski (odcinek trzeci). W taki sposób przesyłany jest jeden bit informacji do WS2812B na 16 kanałach jednocześnie. Stąd wynika natychmiast, że potrzebne są 2 bajty pamięci RAM do przekazania 1 bitu informacji do wszystkich 16 linii sterujących diodami. Ponieważ każda dioda WS2812B wymaga przesłania 24 bitów informacji, do wysterowania 16 diod (po jednej diodzie na każdej linii sterującej) potrzeba 48 bajtów (lub 24 danych typu uint16). Na wspomniane 24 bity składają się trzy bajty odpowiadające ustawieniom kolorów zielonego, czerwonego i niebieskiego. Jeśli więc każda linia sterująca ma po N diod, potrzeba N*24 danych typu uint16 w pamięci RAM. W programie macierzą tą jest WS2812_IO_framedata[].

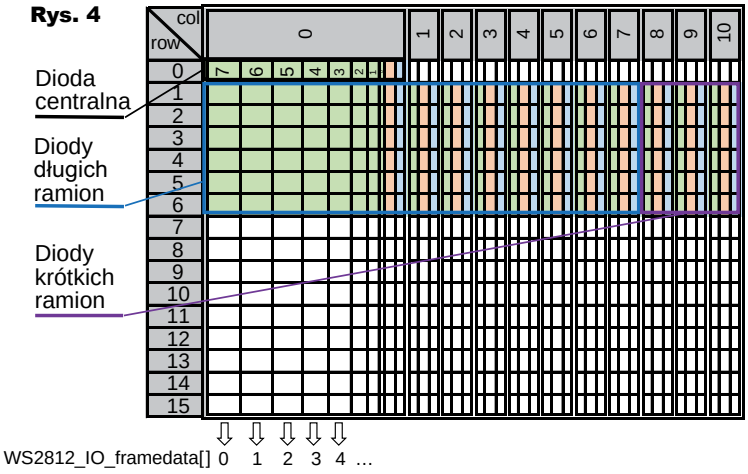
Na koniec, po przesłaniu całego bufora z danymi do diod poprzez port PA, generowane jest przerwanie zakończenia transmisji DMA na kanale 7. W obsłudze tego przerwania włączane jest przerwanie od TIM_IT_Update timera TIM2. W obsłudze tego przerwania odliczana jest liczba przepełnień licznika TIM2 określona parametrem WS2812_DEADPERIOD. Domyślnie jest on ustawiony na wartość 39. Ponieważ TIM2 przepełnia się z okresem 1,25µs (800kHz), oznacza to wygenerowanie 50µs czasu na załadowanie wszystkich przesłanych wartości w wewnętrznym zatrzasku każdej diody. Po tym czasie flaga WS2812_TC ustawiana jest na '1' i diody gotowe są na przyjęcie nowego zestawu danych.

Na **rysunku 4** przedstawiono organizację danych w macierzy WS2812_IO_framedata. Pojedyncza wartość macierzy zawiera pojedynczy bit informacji dla 16 diod na 16 liniach sterujących. Dane zorganizowane są w 8-bitowe informacje, rozpoczynając od najstarszego bitu. Kolory odpowiadają informacji dotyczącej danego koloru diody WS2812 w kolejności zielony, czerwony, niebieski. Parametr col (kolumna) indeksuje kolejną diodę w danej linii row (wiersz). Na rysunku pierwszy bajt każdej z 16 linii jest rozbity na poszczególne bity ponumerowane od 7 do zera (w row=0). Jak wspomniano wcześniej, dioda centralna podłączona jest do linii PA0 (tj. row=0). Na tej linii nie ma więcej diod, więc w tym wierszu używane są 3 bajty, pozostałe 30 bajtów (komórki od col=1 do 10) są nieużywane. Z kolei na liniach PA1-PA6 (co odpowiada row=1...6) znajduje się po 11 diod (8 na długie ramiona i 3 na krótkie). Bajty w macierzy WS2812_IO_framedata oznaczono odpowiednimi ramkami. Bity odpowiadające liniom row=7...15 są nieużywane, więc nie mają znaczenia. W programie stosuję 16-bitową macierz WS2812_IO_framedata, choć dość prosto można zmodyfikować program tak, aby ww. macierz była typu uint8, tym samym zmniejszając ilość zużytej pamięci RAM. W przypadku tej konstrukcji cała macierz zajmuje 528 bajtów i nie jest krytyczne zmniejszanie jej rozmiaru.

W zasadzie na tym się kończy cała trudność programu. Wystarczy teraz odpowiednio modyfikować zawartość macierzy WS2812_IO_framedata tak, aby uzyskać pożądany efekt świetlny. Należy tu dodać, że rozpoczęcie wysyłania informacji do diod na wszystkich liniach rozpoczyna się programowo przez wywołanie funkcji WS2812_sendbuf(uint32_t buffersize). Parametrem tej funkcji jest wielkość macierzy WS2812_IO_framedata, czyli w tym przypadku będzie to 11*24 = 264 danych typu uint16. Zakończenie transmisji danych do diod sygnalizowane jest przez ustawienie flagi WS2812_TC. Zatem przed wywołaniem funkcji WS2812_sendbuf należy zaczekać do zakończenia poprzedniej transmisji.

W celu wprowadzania danych do macierzy WS2812_IO_framedata utworzono szereg wspierających funkcji przyjaznych dla programisty. Zebrano je w **tabeli 1**.

Rys. 4



Program główny, czyli iluminacja gwiazdki, jest niezwykle prosty i opiera się na sekwencyjnym wywoływaniu funkcji generującej odpowiedni schemat iluminacji. W aktualnym programie zaimplementowano kilkanaście różnych schematów. Każdy schemat opakowany jest w szablon funkcji gwiazdka_schemat_X, która przyjmuje dwa parametry: *cykle* oraz *szybkosc* (**listing 1**). Parametr *cykle* odpowiada za liczbę powtórzeń danego schematu, natomiast *szybkosc* umożliwia przyspieszanie/spowalnianie efektu. Szybkość dobiera się doświadczalnie według własnych upodobań.

Warto w tym miejscu wspomnieć o definicji stałej o nazwie GLOBAL_BRIGHTNESS. Łatwo się domyślić, że chodzi o globalną jasność gwiazdki. Domyślnie wartość ta ustawiona jest na 128 (z przedziału 0...255), zatem w połowie zakresu jasności. Stałą tę stosuje się w funkcjach z rozszerzeniem _HSV i przypisuje do parametru value. Nawet przy tym nastawieniu jasność gwiazdy jest bardzo wysoka, a dzięki temu redukujemy maksymalny pobór prądu o połowę. Można próbować z jeszcze mniejszymi wartościami tak, aby zminimalizować temperaturę radiatora stabilizatora LM1084.

Wyjaśnienia wymaga jeszcze schemat 7 (funkcja gwiazdka_schemat_7). W zamyśle ma to być efekt ognia. Iluminacja ta bazuje na kodzie Fire2012 autorstwa Marka Kriegsmana i jest efektem jednowymiarowym, tj. przeznaczona dla linijki diod. Opis kodu źródłowego sugeruje, że najlepsze efekty uzyskuje się przy liczbie diod 30...100. W gwiazdce efekt ten zrealizowano przez wprowadzenie efektu ognia od centralnej diody na zewnątrz każdego z ramion. Należy tu wspomnieć o pewnej wadzie użytego kodu. Zdarza się, że wzorzec mrugania imitujący iskry i płomienie nie jest wystarczająco losowy, co objawia się wyraźnie widocznym cyklicznym mruganiem. Okazjonalnie zdarza się też, że cały wzorzec nie mruga – efekt jest jakby zamrożony. Prawdopodobnie ma to związek z niezbyt dobrym generatorem liczb losowych zawartym w funkcji *lcg_parkmiller*, którego realizację zaczerpnąłem z Wikipedii (hasło: Generator Lehmera). Nie zgłębiałem jednak tej przypadłości, jako że występuje niezbyt często.

Funkcja	Parametry	Opis
WS2812_framedata_setPixel	row, column, red, green, blue	ustawia kolor pojedynczej diody. Każdy kolor wprowadzany jest jako odrębny parametr
WS2812_framedata_setPixelRGB	row, column, rgb	ustawia kolor pojedynczej diody przez podanie wartości rgb typu uint32 w postaci 0x0RGB.
WS2812_framedata_setPixel_HSV	row, column, hue, sat, val	ustawia kolor pojedynczej diody w systemie HSV
WS2812_star_setLongArm_HSV	arm, hue, sat, val	ustawia kolor długiego ramienia *arm* w systemie HSV. *arm* przybiera wartości 0...6. *arm*=0 oznacza diodę centralną.
WS2812_star_setShortArm_HSV	arm, hue, sat, val	ustawia kolor krótkiego ramienia *arm* w systemie HSV. *arm* przybiera wartości 0...6. *arm*=0 oznacza diodę centralną.
WS2812_star_setArm_HSV	arm, hue, sat, val	ustawia kolor dowolnego ramienia *arm* w systemie HSV. *arm* przybiera wartości 0...12. *arm*=0 oznacza diodę centralną.
WS2812_star_setLongRadius_HSV	radius, hue, sat, val	ustawia kolor diod w długich ramionach w danej odległości od centrum. *radius* może przyjmować wartości 0...8, gdzie *radius*=0 oznacza diodę centralną
WS2812_star_setShortRadius_HSV	radius, hue, sat, val	ustawia kolor diod w krótkich ramionach w danej odległości od centrum. *radius* może przyjmować wartości 0...3, gdzie *radius*=0 oznacza diodę centralną

Tabela 1

Montaż i uruchomienie

Jak wspomniano wcześniej, gwiazdka zbudowana jest z dwóch płytek PCB, pokazanych na **rysunku 5** i na **fotografii 1**. Górna płytka zawiera diody w obudowie 5050 oraz kondensatory odsprzęgające w obudowie SMD 0805. Ich montaż nie jest zbyt skomplikowany i nie wymaga komentarza. Dość powiedzieć, że 9-letni syn autora radził sobie z montażem tych elementów nad wyraz sprawnie :)

```
void gwiazdka_schemat_X(uint32_t cykle, uint32_t szybkosc)
{
   uint32_t i, hue, sat, val;

  //ilosc powtorzen danego programu
    while(cykle--)
    {
      while (!WS2812_TC);

      //tu wprowadz nowy wzor swiecenia do
      //macierzy WS2812_IO_framedata

      WS2812_sendbuf( WS2812_BUFFER_SIZE );

      //jakies opoznienie na zaobserwowanie efektu
      Delay( (uint32_t) (1<< 24) >> szybkosc);
    }
}
```

Listing 1

Na końcu wlutowujemy od dołu złącza typu *goldpin*. Posłużą one do połączenia obu płytek. Montaż płytki dolnej najlepiej rozpocząć od montażu stabilizatorów napięć i elementów im towarzyszących. Uwaga: pola oznaczone jako Z1–Z2 są przeznaczone dla zwory, którą montujemy w przypadku LM1084 o stałym napięciu wyjściowym 5,0V. Wówczas nie należy montować elementów R6–R9. Jeżeli zastosowany jest stabilizator LM1084–ADJ, zwory Z1–Z2 nie montujemy. Oczywiście wówczas należy wlutować odpowiednie rezystory R6–R9.

Dlaczego cztery rezystory?

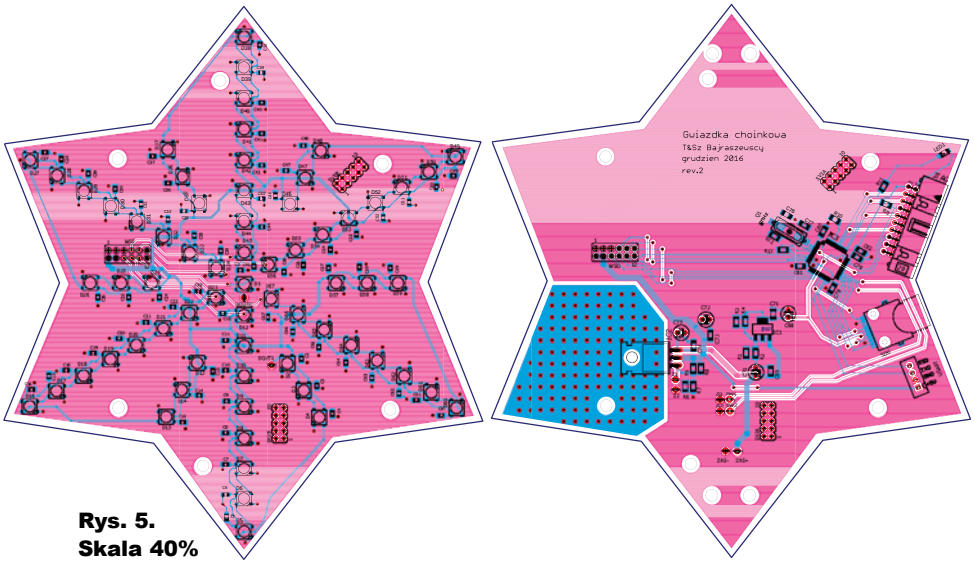
Ułatwia to odpowiednie dobranie wartości rezystorów w celu ustalenia właściwego napięcia wyjściowego. I tak, dla rezystorów R8, R9 o wartości 240 Ohm oraz R7=300 Ohm przy braku rezystora R6 uzyskamy napięcie równe 4,4V. Dla stabilizatora zasilającego mikroprocesor napięcie zasilające ustalono na poziomie 3,6V, które uzyskamy dla następujących wartości rezystorów R1–R4: R1=brak,

R2=402 Ohm, R3=R4=430 Ohm. Tak dobrane wartości zapewniają uzyskanie napięć zasilających, które zapewnią prawidłowe działanie gwiazdki przy skrajnych rozrzutach parametrów elektrycznych diod i mikroprocesora. W praktyce takie skrajne parametry rzadko się zdarzają, więc w celach hobbystycznych z tą precyzją doboru napięć nie należy przesadzać. W prototypie pokazanym na fotografiach zastosowano stabilizator LM1084 o stałym napięciu 5,0V i układ działał prawidłowo przez całe święta. Tak więc, jeśli ktoś nie ma rezystorów o podanych wartościach, może dobrać inne wartości zgodnie ze wzorem $U_{wyj} = 1,25*(1 + R_A/R_B)$, gdzie R_A jest wynikową rezystancją równolegle połączonych rezystorów R1 i R2 (dla IC1) lub R6 i R7 (IC2), a R_B analogicznie dla rezystorów R3 i R4 (IC1) oraz R8 i R9 (IC2).

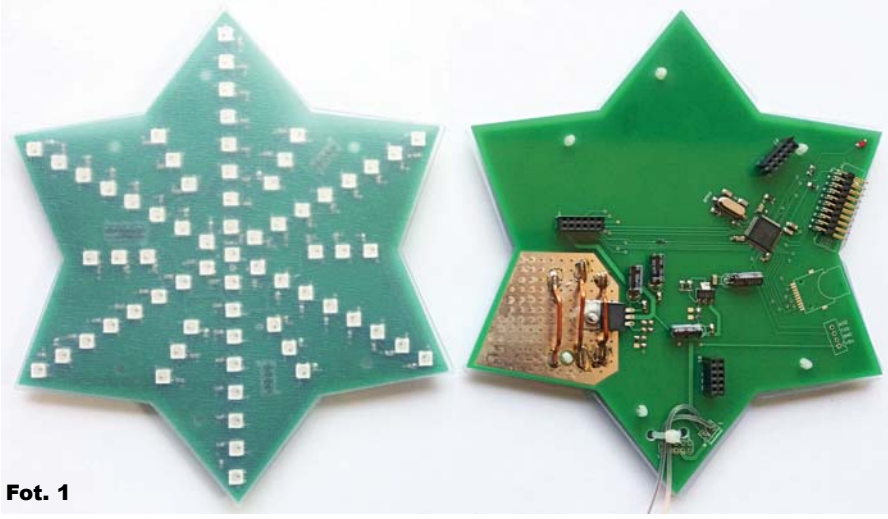
Po przylutowaniu obu stabilizatorów i elementów towarzyszących warto podłączyć napięcie wejściowe i sprawdzić napięcia na wyjściach stabilizatorów. Jeśli napięcia są prawidłowe, można wlutować pozostałe elementy. Na samym końcu wlutowujemy wszystkie złącza dwurzędowe. Złącze JTAG do programowania procesora powinno być kątowe (może to być dwurzędowy *goldpin* zamiast złącza 20-pinowego). Ułatwia to wprowadzanie modyfikacji do oprogramowania.

Prawidłowo zmontowana gwiazdka, po zaprogramowaniu mikroprocesora, powinna od razu działać i cieszyć oko. Przy aktualnym oprogramowaniu stabilizator LM1084 nagrzewa się do ok. 45 °C, co jest jak najbardziej akceptowalne i niegroźne. Niemniej w celu poprawy oddawania ciepła zamontowano trzy grube kawałki przewodu miedzianego, które tworzą rodzaj żeber.

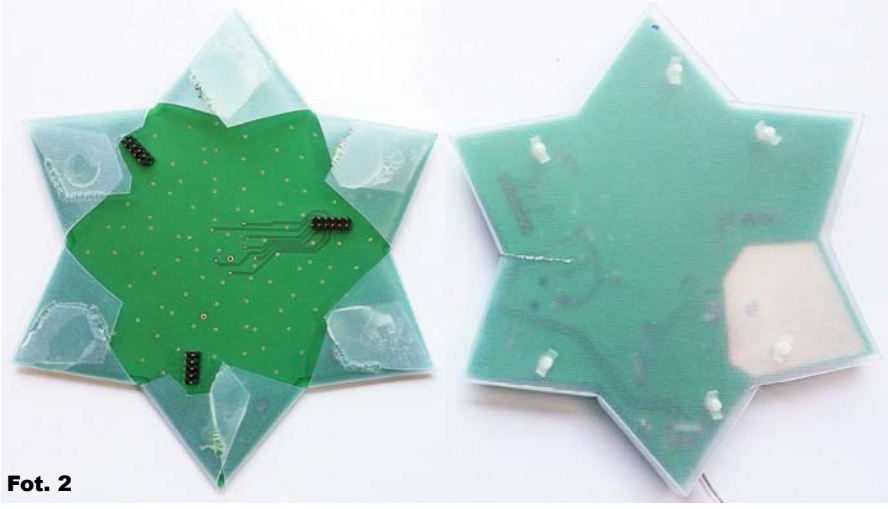
Na koniec warto też zadbać o obudowę gwiazdki. Obudowę wykonano z teczki do dokumentów wykonanej ze sztywnej matowej folii, zakupionej w jednym z popularnych hipermarketów w dziale biurowym. Z folii wycięto dwie gwiazdki z odpowiednio długimi „językami" na każdej krawędzi (**fotografia 2**). W przypadku płytki z diodami języki są na tyle długie, aby po zagięciu nachodziły na siebie, co umożliwi ich połączenie. Niestety, ani klej na gorąco, ani poxipol nie były skuteczne – po pewnym czasie języki się rozklejały. Zastosowałem więc metodę brutalną, acz skuteczną: przygrzewanie lutownicą kolbową. Efekt końcowy można zobaczyć na fotografii 2. Dla dolnej płytki gwiazdka z folii ma języki niezbyt szerokie, rzędu 5mm. Jej montaż do dolnej płytki wykonany jest przez plastikowe klipsy mocujące stosowane zwykle do połączeń mechanicznych między płytkami PCB. W tym celu w dolnej płytce PCB znajduje się sześć otworów o średnicy 3,5mm. Identyczne otwory należy nawiercić w gwiazdce z folii. W ten sposób montaż obudowy z folii do płytki PCB jest trwały, a w razie potrzeby łatwy w demontażu. Wąskie języki przy dolnej płytce zaginamy pod kątem 90°. Tak przygotowaną gwiazdkę można już zamontować na choinkę, ku radości całej rodziny.



Rys. 5.
Skala 40%



Fot. 1



Fot. 2

Fot. 3


Fot. 4

Do zasilania gwiazdki zastosowano 12V zasilacz impulsowy o mocy 50W przeznaczony do zasilania taśm LED (**fotografia 3**), dostępny w znanych hipermarketach ogrodniczych lub narzędziowych. Napięcie wyjściowe zostało wyregulowane dostępnym w zasilaczu potencjometrem na najniższą wartość (10,5V) w celu minimalizacji strat mocy. Zasilacz został zamknięty w obudowie, w której zamontowano włącznik i diodę podłączoną do wyjścia zasilacza (**fotografia 4**). Przewód połączeniowy między gwiazdką a zasilaczem wykonano z kabla antenowego UKF z oryginalną wtyczką. Dzięki temu rozwiązanie jest estetyczne i daje możliwość odłączenia gwiazdki od zasilacza, co jest pomocne podczas montażu gwiazdki na choince.

Możliwości zmian

Jeśli chodzi o schematy iluminacji, możliwości zmian jest bardzo dużo. W obecnym kodzie jest kilkanaście schematów iluminacji. Ilość schematów jest w zasadzie ograniczona ilością kodu pamięci. I tu mała niespodzianka dla lubiących wyzwania. Na płytce sterownika przewidziano możliwość podłączenia karty mikroSD, którą można wykorzystać do zapisu schematów iluminacji. Daje to ogromne możliwości, gdyż można napisać prosty program odczytujący z karty mikroSD wartości będące wartościami macierzy WS2812_IO_frame-data, a całą „inteligencję" iluminacji przenieść do komputera PC. Wymagałoby to oczywiście napisania odpowiedniej aplikacji do generowania schematów iluminacji, ale nie powinno być to skomplikowane zadanie nawet dla średnio zaawansowanego programisty. Na płytce przewidziano też podłączenie dwóch przycisków (pola SW1 i SW2). W obecnym prototypie przyciski są nieużywane, ale można je wykorzystać w różnoraki sposób, np. do sterowania jasnością całej gwiazdki lub do wyboru schematu/programu świecenia.

Cała dokumentacja (kod źródłowe, schemat ideowy i płytki w formacie EAGLE) dostępne są na Elportalu.

Tomasz i Szymon Bajraszewscy
tobajer@poczta.onet.pl

Wykaz elementów

C1-C67	100nF/0805
C69, C71, C73, C74, C78-81	100nF/1206
C66, C67	33pF/1206
C82	4,7uF/25V
R1-R4, R6-R9	opis w tekście
R11, R12	430Ω/1206
R5, R10, R13	10kΩ/1206
D1-D67	WS2812B
LED1	dow. LED/1206
Q1	8MHz/HC49UP
IC1	LM317/SOT223
IC2	LM1084-ADJ
IC3	STM32F103RB
JTAG	złącze męski kątowe 2x10 goldpin
SV1B, SV2B	złącza męskie goldpin dwurzędowe 2x5
SV1A, SV2A	złącza żeńskie dwurzędowe 2x5
WSO	złącze żeńskie dwurzędowe 2x6
WSI	złącze męskie goldpin dwurzędowe 2x6

Płytka drukowana jest dostępna w Sklepie AVT jako AVT3185