

# PWMLEDz: 10-kanalowy sterownik taśm LED z interfejsem Modbus lub SPPoB

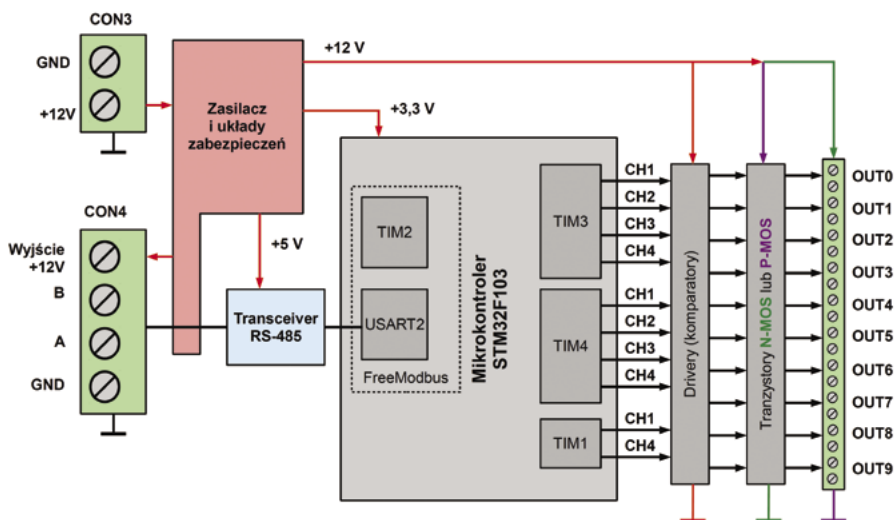
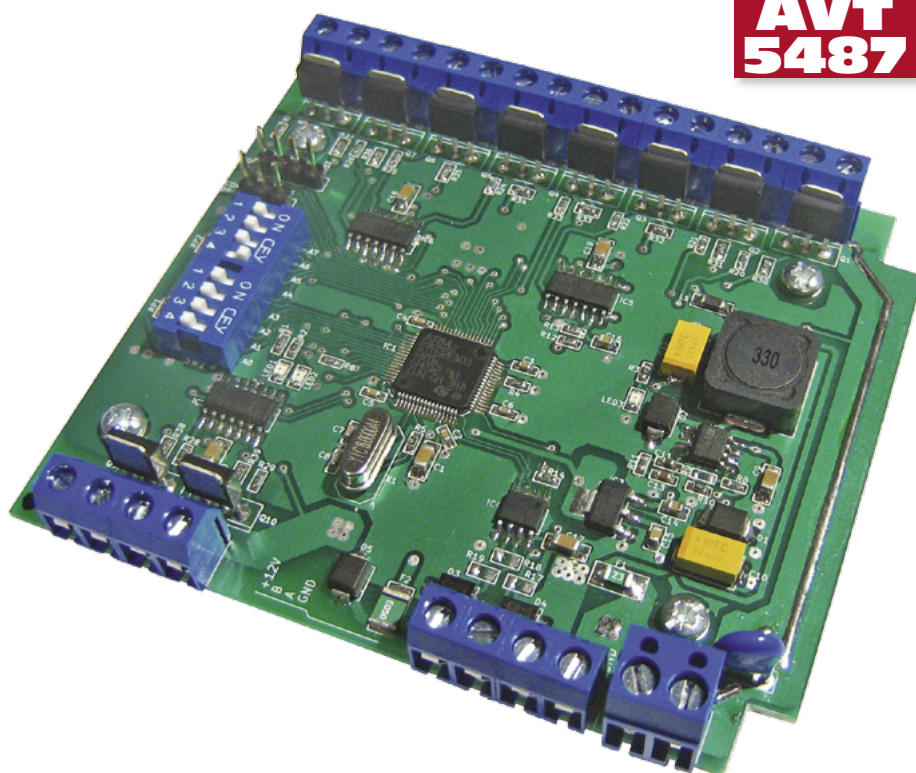
**AVT  
5487**

W artykule zaprezentowano sterownik popularnych taśm LED zasilanych napięciem 12 V. Sterownik został zaprojektowany w dwóch wersjach sprzętowych:

z kluczkowaniem po stronie napięcia dodatniego lub po stronie masy. Ta druga opcja jest przydatna do taśm wielokolorowych, głównie RGB. Obie wersje sprzętu są w pełni kompatybilne programowo, czyli ten sam kod wykonywalny ma działać identycznie na obu wersjach. Oprócz tego, dostępne są także dwie wersje oprogramowania, dzięki czemu sterownik może działać z popularnym protokołem Modbus lub opracowanym przez autora eksperymentalnym protokołem SPPoB opisanym w EP 10/2014.

**Rekomendacje:** urządzenie przyda się do tworzenia unikatowych systemów oświetlenia np. w budynkach.

Wraz z rozwojem technologii, oświetlenie LED zyskuje coraz większą popularność. Diody LED dużej mocy (kilku- lub kilkunastowatowe) dla osiągnięcia optymalnych warunków pracy wymagają stosowania zasilaczy stałoprądowych, które nie są tak proste w konstrukcji i tak powszechne jak stałonapięciowe. Ponadto duża moc skupiona w jednym punkcie wymusza konieczność stosowania specjalnych płytek drukowanych na podłożu aluminiowym pełniących rolę radiatorów, do których takie diody powinny być lutowane. Znacznie prostsze w stosowaniu są elastyczne taśmy z diodami LED zasilane stałym napięciem 12 V. Ich głównymi zaletami jest łatwość obsługi od strony



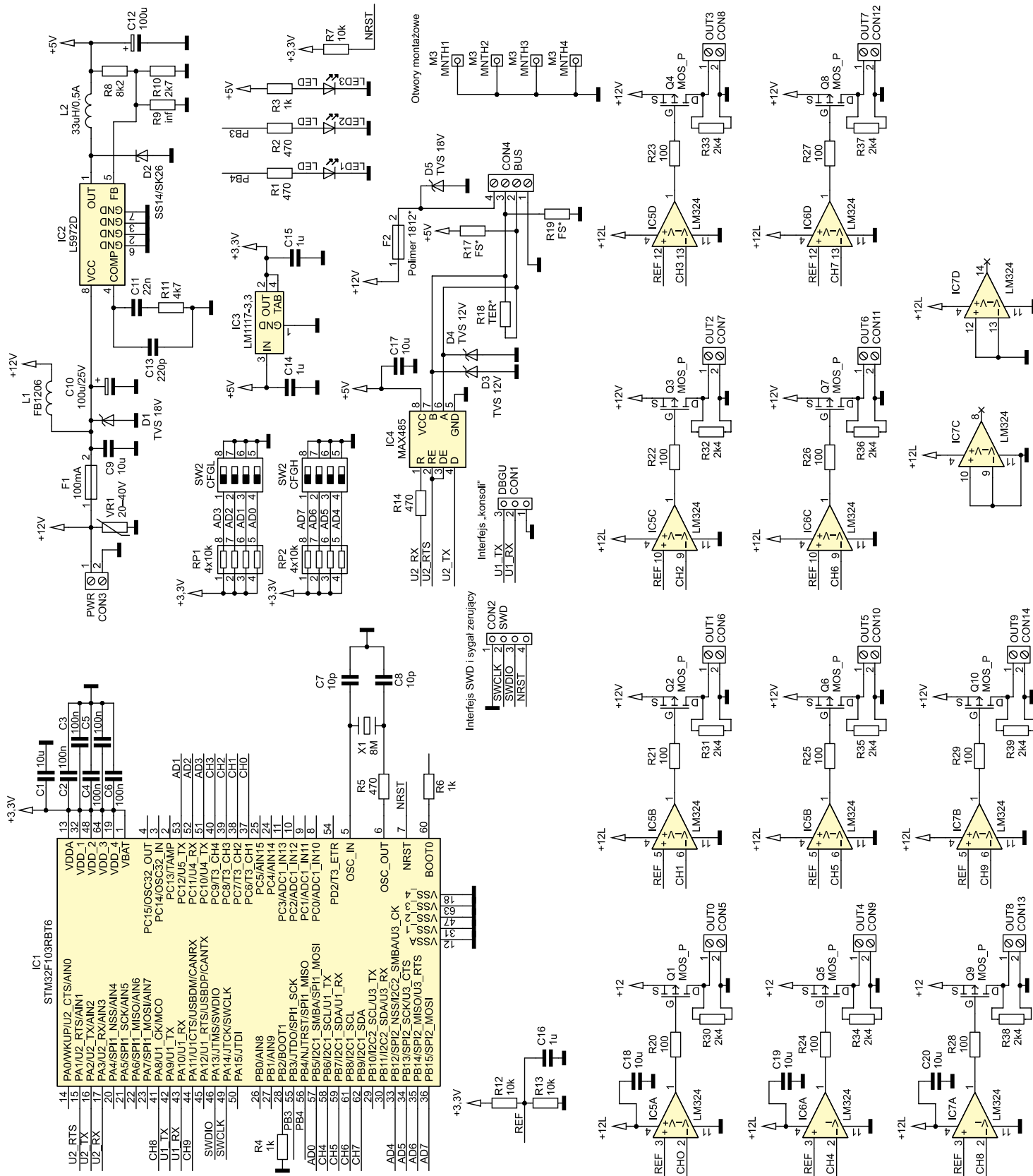
Rysunek 1. Schemat blokowy sterownika taśm LED

elektrycznej, możliwość estetycznego ukrycia dzięki elastycznej i podłużnej formie oraz uzyskiwanie oświetlenia bez tak wyraźnej granicy cienia, jak w przypadku źródeł punktowych.

Aby taśma LED po prostu zaświeciła, potrzebujemy wyłącznie 12-woltowego zasilacza lub dowolnego innego źródła napięcia

o tej wartości. Idąc dalej, zamiast po prostu włączać taśmę LED możemy ustawiać różny poziom jej jasności lub dobrać kolor oświetlenia w zależności od nastroju. Wystarczy do tego odpowiedni sterownik i – dla uzyskania różnych barw światła – taśmy trójkolorowe z diodami LED RGB: czerwonymi, zielonymi i niebieskimi.

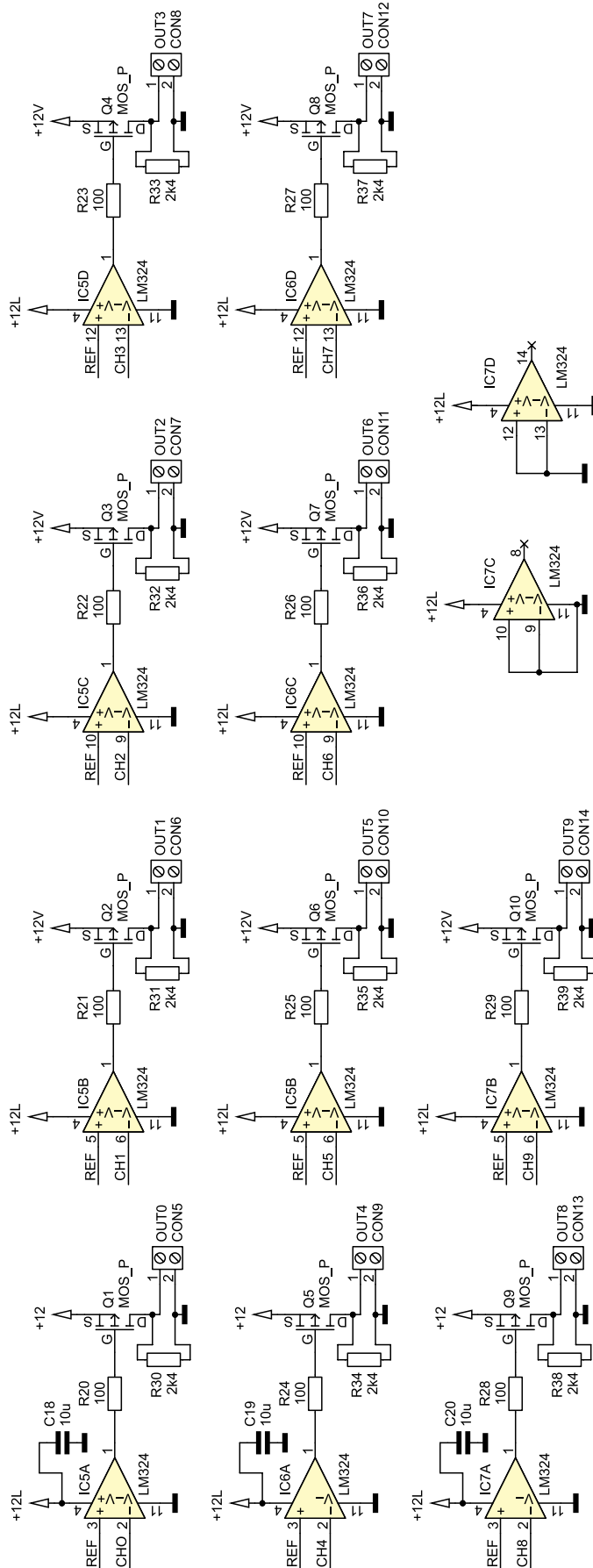
Oczywiście, już od dawna istnieją urządzenia sterujące taśmami LED jedno- i wielokolorowymi. Najczęściej są one kontrolowane z dołączonych pilotów na podczerwień lub czasem także radiowych. Jednak mało jest rozwiązań, w których można jednym pilotem sterować kilka różnych taśm. Podobnie jest z możliwością sterowania wielu pasków LED



Rysunek 2. Schemat ideowy sterownika taśm LED w wersji z tranzystorami P-MOS

przy pomocy komputera lub własnego urządzenia nadrzędnego przez jakiś standardowy interfejs. Wreszcie funkcjonalność dostępnych

rozwiązań, szczególnie tych zdalnie sterowanych, jest narzucona przez producenta i raczej nie będziemy mieć możliwości jej modyfikacji.



Rysunek 3. Schemat ideowy bloku wyjściowego sterownika taśm LED w wersji z tranzystorami N-MOS. Główna część schematu jest identyczna jak na rysunku 2

**W ofercie AVT\***  
**AVT-5487 A**

**Podstawowe informacje:**

- Obsługuje popularne taśmy LED zasilane napięciem 12 V.
- Nominalne napięcie zasilania 12 V DC.
- Pobór prądu bez obciążenia: mniejszy niż 30 mA.
- Sterowanie jasnością przez modulację szerokości impulsów (PWM).
- Łatwa w modyfikacji kompensacja nieliniowości postrzegania jasności.
- Częstotliwość pracy: 550 Hz.
- Testowana obciążalność każdego kanału: 5 A.
- Płytkę pasuje do obudowy Z-101 na szynę DIN.
- Interfejs: protokół Modbus lub SPPoB.
- Oprogramowanie w języku C dla kompilatorów GCC.

**Dodatkowe materiały na FTP:**  
[ftp://ep.com.pl, user: 32086, pass: sqz8sawb](http://ftp://ep.com.pl, user: 32086, pass: sqz8sawb)

• wzory płytek PCB

**Projekty pokrewne na FTP:**  
 (wymienione artykuły są w całości dostępne na FTP)

- AVT-1800 LED Dimmer – regulator oświetlenia LED (EP 5/2014)
- AVT-5400 DMX Dimmer & Relay – regulator oświetlenia i wyłącznik z interfejsem DMX (EP 6/2013)
- AVT-5376 RadioDimmer – regulator oświetlenia w mieszkaniu (EP 1/2013)
- AVT-5361 4dimmer – 4-kanalowy regulator oświetlenia (EP 9/2012)
- AVT-5336 Sterownik oświetlenia sufitu (EP 3/2012)
- AVT-1545 Programowany sterownik świateł (EP 10/2009)
- AVT-5181 Sześciokanałowy dimmer z DMX512 (EP 4/2009)
- AVT-2794 Automatyczny sterownik oświetlenia (EdW 8/2006)
- AVT-924 Programowany sterownik świateł (EP 4/2006)
- AVT-2749 4-kanalowy regulator oświetlenia (EdW 3/2005)
- 12-kanalowy regulator mocy sterowany sygnałem DMX512 (EP 4-5/2003)
- Projekt 089 Zdalnie sterowany regulator oświetlenia (EP 8/2001)
- Projekt 051 Uniwersalny sterownik oświetlenia dyskotekowego (EP 9/1998)
- AVT-445 Inteligentny sterownik oświetlenia (EP 6/1998)
- AVT-1133 Inteligentny regulator oświetlenia (EP 12/1997)

\* Uwaga:  
 Zestawy AVT mogą występować w następujących wersjach:  
 AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
 AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
 AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
 AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf  
 AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf  
 AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

REKLAMA

Projekty na 000

# STM32

[www.stm32.eu](http://www.stm32.eu)

life.augmented

Proponowane rozwiązanie ma na celu ułatwienie konstruktorom uzyskanie urządzenia bardziej elastycznego niż te dostępne w handlu, jednak zachowując istotny parametr, jakim jest niewygórowana cena (choć i tak wyższa niż w przypadku importu z Chin). Ponadto, urządzenie było projektowane tak, aby dało się je zbudować z elementów łatwo dostępnych lub takich, dla których bez problemu da się znaleźć pełnowartościowe zamienniki. Dzięki opisywanemu układowi można zarówno scentralizować jak i zdecentralizować sterowanie paskami LED: uzyskujemy możliwość ustawiania jasności bądź koloru każdego paska z dowolnego miejsca bez fizycznego przenoszenia pilota. Integracja sterownika z innymi urządzeniami domowymi i włącznikami ściennymi bądź zdalnymi także może stać się nieco prostsza niż w przypadku rozwiązań produkowanych masowo. Dopisanie kodu podającego odpowiednie rozkazy do opisywanego sterownika (lub wielu sterowników) jest bowiem całkiem prostym zadaniem dla Czytelników mających choćby odrobinę doświadczenia z programowaniem interfejsu szeregowego w mikrokontrolerach lub na komputerze. Możemy wreszcie dowolnie modyfikować funkcjonalność sterownika taśm dysponując wyłącznie darmowymi narzędziami programowymi oraz niedrogim, ogólnodostępnym programatorem.

**Sprzęt**

Schemat blokowy sterownika przedstawiono na **rysunku 1**. Urządzenie zostało zaprojektowane w dwóch prawie identycznych (może lepsze słowo to „komplementarnych”) wersjach. Centralnym elementem jest oczywiście mikrokontroler, tutaj STM32F103 (testy były przeprowadzane na modelu STM32F103RBT6). Wybór mikrokontrolera był podyktowany jego popularnością, dobrym stosunkiem funkcjonalności do ceny, a przede wszystkim ilością dostępnych timerów z możliwością generowania sygnałów o modulowanej szerokości impulsów (PWM). Na rys. 1 zaznaczono, z których timerów (moduły TIM) korzystają poszczególne kanały wyjściowe OUT0...OUT9. Dodatkowo, w wersji firmware'u z protokołem Modbus potrzebny jest kolejny timer, tutaj TIM2, natomiast nie ma potrzeby używania go w wersji z protokołem SPPoB.

Mamy też dodatkowe wyjście zasilania na złączu CON4. Jeśli zaplanujemy stosowanie innych urządzeń z interfejsem RS-485, to przy okazji dołączenia ich do magistrali warto będzie doprowadzić do nich zasilanie. Zasilanie taśm LED wymaga nie raz całkiem mocnego zasilacza, a w takiej sytuacji łatwo o pewien nadmiar wydajności prądowej. Z tego nadmiaru najczęściej bez problemu można uszczknąć kilka

do kilkuset miliamperów do wygodnego zasilania klawiatur sterujących lub „inteligentnych” czujników automatycznie włączających oświetlenie.

Schemat ideowy sterownika taśm w wersji z kluczowanym napięciem +12 V przedstawiono na **rysunku 2**. Do kluczowania wykorzystuje on tranzystory z kanałem P. Alternatywnie można zbudować wersję z tranzystorami N, w której kluczowana jest masa, a napięcie +12 V doprowadzone jest na stałe. Obie wersje różnią się jedynie fragmentem schematu przedstawionym na **rysunku 3**. Inne są oczywiście tranzystory mocy oraz podłączenie sygnałów do „driverów” ich bramek.

Do budowy prototypu w wersji P-MOS zastosowano tranzystory AP4435GJ-HF-3TB w obudowach TO-251, natomiast dla wersji N-MOS były to AP9962GJ-HF-3TB w takiej samej obudowie. Tranzystory te charakteryzują się niezłymi parametrami elektrycznymi przy niskiej cenie. W przypadku trudności ze zdobyciem wyżej wspomnianych tranzystorów, można śmiało rozważyć zastosowanie zamienników, ponieważ układ sterownika jest mało wrażliwy na ich parametry. Chyba jedynym mniej typowym parametrem zastosowanych tranzystorów jest relatywnie niskie napięcie progowe VGS(th) mieszczące się w przedziale od 1 do 3 V przy sporym maksymalnym prądzie drenu wynoszącym 20 A dla P-MOS i 32 A dla N-MOS (o niskie napięcie VGS(th) łatwiej wśród tranzystorów FET przeznaczonych do małych prądów).

Jeśli popatrzeć na rozwiązanie driverów dla bramek tranzystorów mocy, to może się wydawać, że jest to rozwiązanie co najmniej wątpliwe – bramki są bowiem sterowane przez komparatory zbudowane z jednych z najtańszych i najpopularniejszych wzmacniaczy operacyjnych LM324. Zastosowanie rozwiązania z komparatorami ma jednak kilka zalet. Przede wszystkim nie podwyższa kosztów urządzenia, ponieważ ceny dedykowanych układów driverów są zbliżone do ceny zastosowanych tranzystorów, co przy skali 10 kanałów i kilku sterowników zaczyna generować odczuwalną różnicę. W razie braku wybranego układu drivera trudniej jest znaleźć dla niego pasujący na płytkę zamiennik, podczas gdy zastosowane wzmacniacze operacyjne można nabyć w przysłowiowym sklepie za rogiem. Można było także zaprojektować drugi układ z komplementarnymi tranzystorami działający z identycznym oprogramowaniem (bez konieczności odwracania przebiegów PWM z mikrokontrolera) przez prostą zamianę wejść komparatorów.

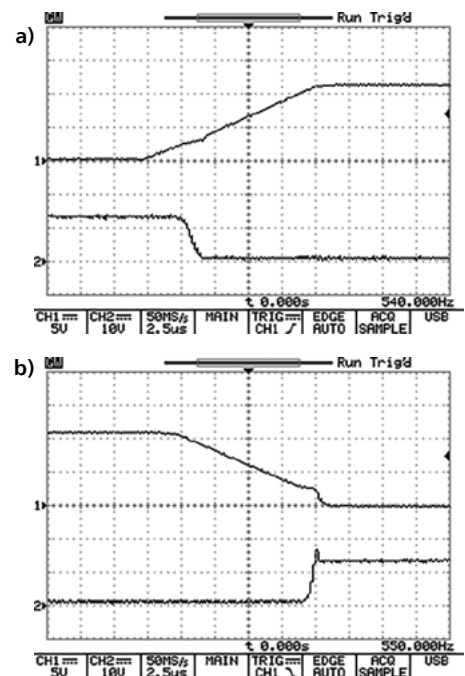
Przy zastosowanym prostym układzie sterowania bramek tranzystorów oraz

częstotliwości przebiegu PWM wynoszącej 550 Hz, tranzystory nie wydzielają zauważalnych ilości ciepła, nawet pod obciążeniem ok. 5 A. Na **rysunku 4** przedstawiono zrzuty ekranu oscyloskopu dla włączania i wyłączania obciążenia w wersji N-MOS. W kanale pierwszym (górny wykres) widzimy napięcie na bramce tranzystora, natomiast w drugim, napięcie na obciążeniu (tutaj prąd po załączeniu to ok. 2 A). Czas narastania napięcia na bramce można szacować na ok. 10 μs, jednak dzięki niskiemu napięciu załączenia VGS(th) tranzystory włączają obciążenie znacznie szybciej, tj. w czasie niewiele przekraczającym 1 μs.

Sumaryczna obciążalność sterownika podyktowana jest głównie ograniczeniami jego płytki drukowanej i zastosowanego złącza zasilania CON3. W egzemplarzach prototypowych zastosowano główne złącze zasilania o maksymalnym prądzie 24 A. Kolejnym ograniczeniem jest maksymalny prąd, który mogą przewodzić obwody drukowane na płytce sterownika – przy dużym obciążeniu ścieżki będą się nagrzewać.

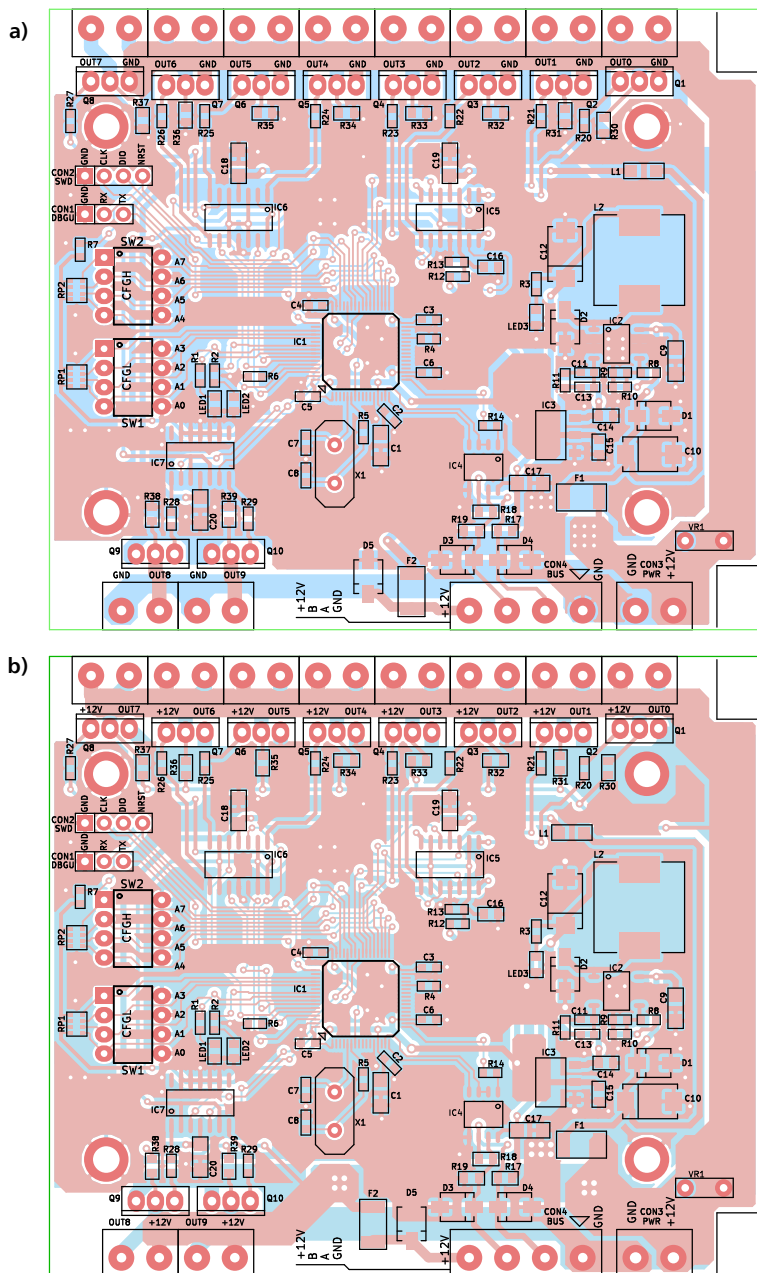
**Zasilanie i zabezpieczenia**

Blok zasilania sterownika taśm LED składa się z dwóch popularnych stabilizatorów i układów zabezpieczających. Napięcie ok. 5 V uzyskuje się z układu IC2 (L5972D). Układ ten właściwie pełni funkcję stabilizatora wstępnego, z którego dalej napięcie jest obniżone do 3,3 V w układzie IC3 (LM1117-3.3). Przy okazji napięcie 5 V wykorzystywane jest do zasilania transceivera RS-485. Dzięki zastosowaniu wstępnego



**Rysunek 4. Przebiegi czasowe prezentujące a) włączanie i b) wyłączanie obciążenia dla wersji NMOS (kanał 1 oscyloskopu: bramka tranzystora, kanał 2: napięcie na obciążeniu)**





Rysunek 5. Schematy montażowe sterownika taśm LED: w wersji z tranzystorami: a) P-MOS, b) N-MOS

stabilizacji na dobrym stabilizatorze impulsowym, nie ma problemu z wydzielaniem dużej ilości energii w postaci ciepła, czego następstwem jest to, że sam sterownik pobiera mało prądu (poniżej 30 mA). Może to mieć znaczenie ekonomiczne, jeśli sterownik taśm ma być cały czas podłączony do zasilania.

Układy zabezpieczeń mogą uratować sterownik i paski LED w razie wystąpienia przypadkowych przepięć lub bardziej pechowej awarii zasilacza. Jednak nie są one konieczne do działania sterownika i można ich albo nie montować (w przypadku diod TVS i warystora), albo zastąpić zworami (bezpieczniki polimerowe). Warystor może być przydatny np. wtedy, gdy planujemy stosować długie przewody pomiędzy zasilaczem a sterownikiem lub „niepewny” zasilacz. Gdy zasilacz znajduje się blisko

urządzenia, można śmiało zrezygnować z warystora. Z racji, że warystor może wydzielać dużo energii w momencie wystąpienia przepięcia, zalecane jest zabezpieczenie całego układu dodatkowo zewnętrznym, szybkim bezpiecznikiem włączonym na linii +12 V pomiędzy sterownikiem a zasilaczem. W przeciwnym wypadku nadmiar energii może spowodować dosłowne spalenie warystora, co w czarniejszym scenariuszu może wywołać poważne niebezpieczeństwo pożaru. Zarówno bezpiecznik jak i warystor należy dobrać w zależności od planowanego maksymalnego prądu zasilania. Napięcie pracy warystora powinno być o co najmniej kilka woltów wyższe niż 12 V, aby nie wywoływać „fałszywych alarmów”, które przyczyniają się do degradacji parametrów warystora. Tak czy inaczej, najbardziej wrażliwe elementy

sterownika zostały zabezpieczone dodatkowo diodą D1. Jako D1 w egzemplarzach testowych zostały zastosowane jednokierunkowe diody TVS o napięciu znamionowym ok. 18 V, model SMBJ18A. Bezpiecznik polimerowy F1 pomaga uchronić diodę D1 przed przegrzaniem w przypadku podawania zbyt wysokiego napięcia na wejście zasilania przez dłuższy czas. Komparatory pracujące jako drivery tranzystorów mocy nie mają własnego lokalnego stabilizatora, dlatego zasilane są przez koralik ferrytowy L1, tłumiący zakłócenia o wyższych częstotliwościach, oraz dodatkowo ich zasilanie jest filtrowane przez kondensatory C18-C20. Te elementy warto zastosować, ponieważ na szynie zasilania +12 V mogą pojawić się zakłócenia wynikające z impulsowego charakteru obciążenia. Wyjście zasilania (na złączu CON4) dla innych urządzeń z interfejsem RS-485 także jest w pewnym stopniu zabezpieczone. Przed skutkami zewnętrznego zwarcia chroni je polimerowy bezpiecznik F2, natomiast dioda TVS D5 ma częściowo chronić sterownik przed przepięciami przenikającymi z wyjścia zasilania.

Przewidziano także miejsce na diody TVS (D3 i D4) dla linii A i B interfejsu RS-485. Diody D3 i D4 nie muszą być specjalizowane dla RS-485 – płytką została zaprojektowana pod popularne obudowy większych diod, DO-214AA. W wykazie elementów uwzględniono diody TVS jednokierunkowe na 12 V (zakładamy, że wszystkie urządzenia będą na wspólnej masie). Z racji jednak, że transceivery RS-485 mogą pracować także w zakresie napięć ujemnych na magistrali, można tutaj zastosować diody dwukierunkowe na nieco niższe napięcie.

### Interfejsy zewnętrzne

Interfejs RS-485 zrealizowany jest standardowo. Z racji, że do zasilania transceivera IC4 przewidziano napięcie 5 V ze stabilizatora o sporej wydajności, można zastosować zarówno leciwy i prądożerny SN75176B, jak też jego nowocześniejsze odpowiedniki, np. MAX485. Wejście USART2\_RX (wyprowadzenie PA3) mikrokontrolera wg dokumentacji producenta nie akceptuje napięcia 5 V, dlatego potrzebny jest co najmniej szeregowy rezystor R14 na wyjściu odbiornika (nóżka 1) w układzie IC4. Wartość 470 Ω dla R14 będzie wystarczająca dla ograniczenia prądu wstrzykiwanego do mikrokontrolera w stanie wysokim (producent zaleca, by prąd ten nie przekraczał 5 mA). Właściwe ograniczenie napięcia następuje w mikrokontrolerze, na wewnętrznych elementach zabezpieczających jego wejścia. W sterowniku taśm możliwe jest zarówno zamontowanie terminatora (R18) jak i rezystorów

fail-safe (R17, R19) dla magistrali RS-485. Wg zasad, terminatory stosujemy na końcach magistrali. Rezystory fail-safe wystarczy umieścić w jednym miejscu magistrali (np. razem z jednym terminatorem). W praktyce przy krótkich magistralach i niskich szybkościach transmisji terminatory mogą nie być potrzebne, natomiast przydatne są rezystory fail-safe, które mogą mieć wartości nawet rzędu kilku kiloomów jeśli nie stosujemy terminatorów.

Jedynymi elementami sterownika przewidzianymi do bezpośredniej interakcji z użytkownikiem (tutaj właściwie serwisantem) są przełączniki typu „dipswitch” SW1 i SW2 oraz diody LED1 i LED2. Dodatkowa dioda LED3 służy do wskazywania obecności napięcia zasilania. Przełączniki SW1 i SW2 są bardzo przydatne do ustawiania adresu urządzenia zarówno w wersji dla protokołu Modbus jak i SPPoB. Trzeba tylko brać pod uwagę, że nie każde ich ustawienie jest poprawne bądź ma sens, np. 0000 0000 dla Modbus czy 1111 1111 dla SPPoB.

## Montaż

Płytką drukowaną sterownika taśm LED została zaprojektowana tak, aby dało się ją umieścić w obudowie typu Z-101 z możliwością montażu na szynie DIN. Schemat montażowy wersji N-MOS jak i P-MOS przedstawiono na **rysunku 5**. Jeśli planujemy umieszczenie płytki w obudowie Z-101, należy wyciąć dwa narożniki z prawej strony płytki wg linii zaznaczonych na warstwie opisowej (najlepiej jest to zrobić jeszcze przed rozpoczęciem montażu elementów elektronicznych). Konieczność samodzielnego wycinania narożników podyktowana jest tym, że wykonanie płytek o nietypowych kształtach bywa droższe niż prostokątnych.

Z racji dużej dopuszczalnej obciążalności poszczególnych kanałów, całe urządzenie może pobierać znaczny prąd. Z tego powodu na płytkach w obu wersjach złącze śrubowe głównego zasilania CON3 jest nieco odsunięte od złącza sąsiedniego CON4, by dało się zastosować większe gabarytowo złącze o dużym maksymalnym prądzie przewodzenia, np. 24 A jak na fotografii 1. Dodatkowo, ścieżki na płytkach drukowanych, które mają przewodzić największy prąd, mają częściowo odsłoniętą warstwę maski lutowniczej, aby można było np. pokryć je grubszą warstwą spoiwa lub nawet wlotować dodatkowy drut zmniejszający ich rezystancję.

Sam montaż elementów raczej nie wymaga dodatkowego komentarza. W wersji okrojonej z zabezpieczeń można pominąć montaż diod TVS i warystora, a zamiast bezpieczników polimerowych zamontować zwory. Zworę można wmontować także

zamiast koralika ferrytowego L1. Więcej szczegółów na temat doboru elementów zabezpieczających znajdziemy w sekcji Zasilanie i zabezpieczenia niniejszego artykułu.

Przed wlutowaniem mikrokontrolera, warto sprawdzić poprawność działania stabilizatorów dostarczających napięcia zasilające +5 V oraz +3,3 V. Napięcie +5 V może być w rzeczywistości nieznacznie niższe z racji, że rezystory R8 i R10 użyte do sprzężenia zwrotnego dobrane są spośród popularnych wartości (nie ma to jednak negatywnego wpływu na pracę układu). Jeśli jednak ktoś chciałby uzyskać dokładniejszy poziom tego napięcia, należy dobrać inne wartości rezystorów R8, R9 i R10 ustalających napięcie wyjściowe IC2.

Po zamontowaniu wszystkich elementów i sprawdzeniu poprawności montażu można zaprogramować mikrokontroler dostarczonym kodem wynikowym, ewentualnie wcześniej samemu skompilować źródła. Do zaprogramowania będzie potrzebny programator z interfejsem SWD, zgodny z ST-Link. Jako programatora można również użyć którejś z płytek z rodziny „Discovery”, wyprowadzając interfejs SWD z ich wbudowanego programatora. W przypadku stosowania programatora ST-Link lub zamiennika, należy zapewnić mu napięcie 3,3 V na jego złączu, choć nie jest ono wystawione na CON2 sterownika taśm. W przypadku programatora ZL30PRGv2 wystarczy zewrzeć jego wyprowadzenia o numerach 1 i 19 na głównym złączu JTAG/SWD. Po poprawnym zaprogramowaniu w obu wersjach sterownika powinna migać dioda LED1.

## Programy sterujące

Programy dla obu wersji zostały napisane w języku C, a dołączone projekty były kompilowane za pomocą kompilatora GCC z pakietu Yagarto. Domyślnie powinny działać polecenia *make all* (kompilacja) i *make clean* (usuwanie plików wynikowych z katalogu projektu).

Programy sterujące nie korzystają z żadnego systemu operacyjnego, a główna pętla programowa znajduje się w plikach *main.c*, w funkcjach *main*. Stamtąd jest wywoływana m.in. funkcja *ledTask*, odpowiedzialna za wpisywanie zadanych z zewnątrz poziomów jasności do poszczególnych kanałów z zapewnieniem ewentualnego płynnego przejścia od jasności aktualnej do zadanej. Funkcja *ledTask* jest prawie identyczna dla wersji z interfejsem Modbus jak i SPPoB – różnice dotyczą głównie organizacji struktur danych, natomiast jej ogólna logika działania jest taka sama.

Sterowniki generatorów PWM wbudowanych w mikrokontroler znajdziemy

w *drivers/pwm.c* (takie same dla obu wersji). Moduł *pwm.c* ma bardzo prosty interfejs składający się z dwóch funkcji widzianych na zewnątrz: *pwmInit* oraz *pwmSet*. W funkcji *pwmInit* konfigurowane są odpowiednie sygnały zegarowe, wszystkie moduły TIM, ich wyjścia OC (Output Compare) oraz aktywowane są alternatywne funkcje odpowiednich wyprowadzeń GPIO. Po przeprowadzeniu inicjalizacji można wywoływać funkcję *pwmSet* o następującej deklaracji: *void pwmSet(uint8\_t channel, uint8\_t value)*. Funkcja przyjmuje dwa dość oczywiste parametry. Pierwszy (*channel*) mówi, do którego kanału w sterowniku chcemy wpisać wartość, natomiast samą wartość jasności przekazujemy w drugim argumencie, *value*. Poprawnie obsługiwany zakres argumentu *channel* to liczby całkowite w zakresie od 0 do 9 odpowiadające kanałom wyjściowym sterownika. W parametrze *value* można przekazywać liczby od 0 do 255.

Liczba zadawanych poziomów jasności jest co prawda 8-bitowa, jednak wyjściowe kanały pracują z rozdzielczością 16-bitową. Przekształcenie liczby wejściowej naysterowanie kanału realizowane jest przy pomocy funkcji kompensującej nieliniowy charakter postrzegania jasności przez ludzki wzrok. Przekształcanie wejściowych liczb 8-bitowych z argumentu *value* na liczby 16-bitowe odbywa poprzez 256-elementową tablicę stałych wartości 16-bitowych o nazwie PWM\_LUT (definicja w pliku *pwm\_lut.h* w katalogach głównych projektów). Liczba *value* będąca argumentem funkcji *pwmSet* używana jest jako indeks tej tablicy. Co prawda moc obliczeniowa i częstotliwość taktowania mikrokontrolera pozwoliłyby na przeprowadzenie nawet dość złożonych obliczeń na bieżąco, to jednak rozwiązanie z tablicą LUT (od *lookup table*) powoduje, że czas przekształcania *value* na współczynnik wypełnienia jest taki sam niezależnie od stopnia złożoności funkcji korekcji, a dodatkowo możemy sobie pozwolić na większą swobodę w doborze samej funkcji.

## Krzywa korekcji

Wzrok i słuch ludzki działają – można by rzec – „w skalach logarytmicznych”. Gdy jest bardzo ciemno, oczy są czułe na najmniejsze zmiany natężenia światła, a kiedy jest cicho, słyszymy najdrobniejsze szmery. Natomiast przy bardzo dużych natężeniach światła lub dźwięku, zauważalne zmiany występują dopiero po wyraźnym zwiększeniu mocy źródła czy to dźwięku, czy światła. Gdyby zatem sterować jasnością diod LED w paskach liniowo, to wydawałoby nam się, że daleko przed połową skali diody mają już pełną jasność, więc jeśli założymy, że postrzeganie ma charakter

	A	B	C	D	E	F	G	H	I	J	K	L
1	value (liczba wejściowa)	value znorm. do zakresu 0...1	FK (Wartość funkcji korekcji)	PWM (F znorm. do 0...1)	PWM_LUT (F znorm do 0-65535)	Znaki formatowania	Indeks w LUT = value (komentarz)		a	b	offset (przesunięcie)	Maksymalna wartość FK (pole robocze)
2	0	0,000	0,000	0,00000	0	,	# 0		1	2,3	0,85	199,376
3	1	0,004	0,871	0,00437	286	,	# 1					
4	2	0,008	0,892	0,00448	293	,	# 2					
5	3	0,012	0,914	0,00459	301	,	# 3					
6	4	0,016	0,937	0,00470	308	,	# 4					
7	5	0,020	0,959	0,00481	315	,	# 5					
8	6	0,024	0,983	0,00493	323	,	# 6					
9	7	0,027	1,006	0,00505	331	,	# 7					
10	8	0,031	1,031	0,00517	339	,	# 8					
11	9	0,035	1,056	0,00529	347	,	# 9					
12	10	0,039	1,081	0,00542	355	,	# 10					
13	11	0,043	1,107	0,00555	364	,	# 11					
14	12	0,047	1,133	0,00568	372	,	# 12					
15	13	0,051	1,160	0,00582	381	,	# 13					
16	14	0,055	1,187	0,00596	390	,	# 14					
17	15	0,059	1,216	0,00610	400	,	# 15					
18	16	0,063	1,244	0,00624	409	,	# 16					
19	17	0,067	1,273	0,00639	419	,	# 17					
20	18	0,071	1,303	0,00654	428	,	# 18					
21	19	0,075	1,334	0,00669	438	,	# 19					
22	20	0,078	1,365	0,00685	449	,	# 20					
			1,397	0,00701	459	,	# 21					

Krzywa korekcji PWM(value)



Rysunek 6. Nagłówek arkusza pwm\_lut.ods służącego do generowania krzywych korekcji współczynnika wypełnienia wyjściowych przebiegów PWM

Tabela 1. Mapa rejestrów sterownika taśm LED w wersji obsługującej protokół Modbus

Jasność		Czas zmiany jasności (przejścia)	
Adres	Kanał	Adres	Kanał
101	0	201	0
102	1	202	1
103	2	203	2
104	3	204	3
105	4	205	4
106	5	206	5
107	6	207	6
108	7	208	7
109	8	209	8
110	9	210	9

logarytmiczny, to jego kompensacja może być przeprowadzona przez działanie odwrotne, czyli przez zastosowanie funkcji wykładniczej. Takie przekształcenie może zostać zrealizowane w tablicy PWM\_LUT sterownika taśm LED.

Liczby wyjściowe w tablicy PWM\_LUT oczywiście nie muszą być dobierane i wpisywane ręcznie każda z osobna. Może je za nas wygenerować prosty arkusz kalkulacyjny, z którego tylko skopiujemy wyniki. Taki arkusz z przykładową krzywą korekcji został dołączony do materiałów do niniejszego artykułu i znajduje się w pliku **pwm\_lut.ods**. Z pewnością może on ułatwić dopasowanie funkcji korekcji jasności do posiadanych pasków LED lub po prostu wedle własnego uznania. Jest on opracowany w programie Calc z pakietu LibreOffice, aby dało się bez trudu i za darmo uzyskać możliwość jego otwarcia i edycji dla wielu systemów operacyjnych.

Zrzut nagłówka i kilku początkowych wierszy tego arkusza przedstawiono na **rysunku 6**. Kolejne liczby wejściowe (te, które będziemy zadawać w argumencie value) znajdują się w kolumnie A. Dla ułatwienia liczby te są znormalizowane do zakresu od 0 do 1 (kolumna B) i dopiero w takiej postaci poddawane są przekształceniu przez funkcję korekcji. Dla niezerowych indeksów value funkcja korekcji FK zadana jest następującą zależnością  $FK = a \cdot (10b \cdot value - 1) + offset$ , natomiast dla indeksu value wynoszącego 0 ma ona wartość 0. Parametr a w funkcji FK możemy zadawać w komórce I2 dostarczonego arkusza i może on być używany opcjonalnie do pionowego skalowania krzywej (jeśli nie jest potrzebny, ustawiamy go na 1). Natomiast parametr b zadawany w komórce J2 decyduje o tym, jak mocno „zagięta” jest funkcja FK (wystarczy wykonać parę prób na arkuszu, aby zobaczyć o co chodzi). Dodatkowy parametr offset zadawany w K2 służy do zapobiegania pustym wartościom na początku skali. Bez niego początek skali zawierałby liczby generujące tak krótkie impulsy, że nie pojawiałyby się one na wyjściach sterownika. Odpowiednio dobrany (choćby eksperymentalnie) parametr offset spowoduje, że każda niezerowa liczba w value spowoduje jakieś wystrojenie diod LED w paskach. Pojawiająca się w komórce L2 maksymalna wartość FK osiągnięta w kolumnie C pozwala znormalizować wyjściową wartość funkcji FK do zakresu 0...1, a dalej na liczbę 16-bitową z zakresu 0...65535. Wyróżnioną na kolor jasnoniebieski zawartość kolumn E, F i G

wystarczy zaznaczyć, skopiować i wkleić do wnętrza tablicy PWM\_LUT w pliku **pwm\_lut.h**. Oczywiście właściwe wartości są w kolumnie E, natomiast reszta (F i G) to przecinki wymagane przez składnię języka C i nieobowiązkowe komentarze zawierające numer indeksu w tablicy PWM\_LUT dla każdej wartości.

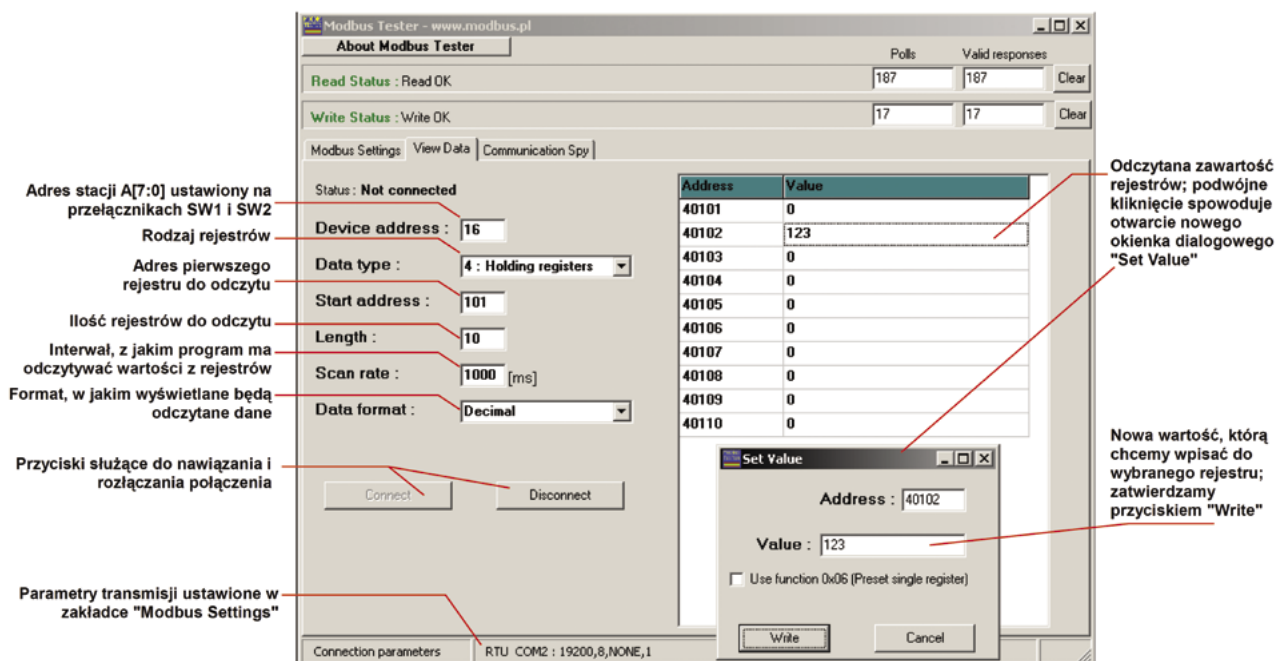
W indywidualnym doborze preferowanej funkcji korekcji pomocne może być zaobserwowanie trwającego ok. 3...4 sekundy, płynnego przejścia przez wszystkie wartości jasności (liniowa zmiana value od 255 do 0) i ocenienie subiektywnego wrażenia „liniowości” zmian. Nie powinno się tej funkcji dobierać na podstawie bezpośredniej, bliskiej obserwacji diod LED w pasku, ponieważ jasne punktowe źródła światła mogą wywoływać chwilowe oślepienie. Lepiej będzie przykryć testową taśmą LED zwykłą, białą kartką papieru dla uzyskania światła rozproszonego i w ten sposób obserwować przejścia jasności.

REKLAMA

Projekty na...  
**STM32**

[www.stm32.eu](http://www.stm32.eu)

**ST** life.augmented  
**KAMAMI**



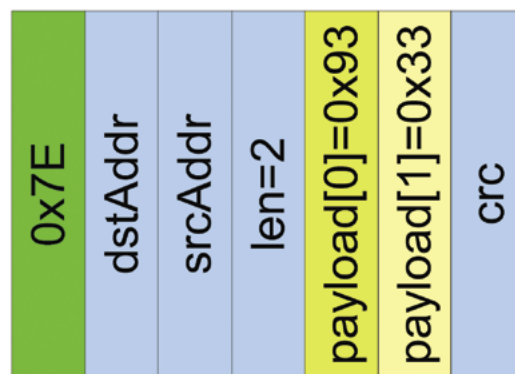
Rysunek 7. Widok okna programu Modbus Tester z wprowadzonymi ustawieniami do testowania zapisu i odczytu rejestrów sterownika

### Interfejs zewnętrzny – protokół Modbus

Wersja oprogramowania komunikująca się przez interfejs Modbus korzysta z popularnej i łatwiej w obsłudze biblioteki FreeModbus dostępnej na stronie <http://goo.gl/JGDrtE>. Przy domyślnych ustawieniach programu korzystamy z Modbus RTU, czyli binarnej odmiany protokołu. Parametry transmisji są następujące: 19200 baud, 8 bitów danych, 1 bit stopu, brak bitu parzystości.

Adres urządzenia (stacji) Modbus ustawiamy na przełącznikach SW1 i SW2 w zakresie od 1 do 247. Mapę rejestrów wewnętrznych sterownika tąm w wersji z protokołem Modbus przedstawiono w tabeli 1. Dla uproszczenia wszystkie zastosowane rejestry są typu holding register. Mamy zarówno możliwość wpisu wartości do poszczególnych rejestrów, jak i ich odczytu. Rejestry mają rozmiar 16 bitów, jednak wpisy do nich są ograniczone do liczby 255. Wpisanie większej liczby spowoduje jej automatyczną redukcję do 255.

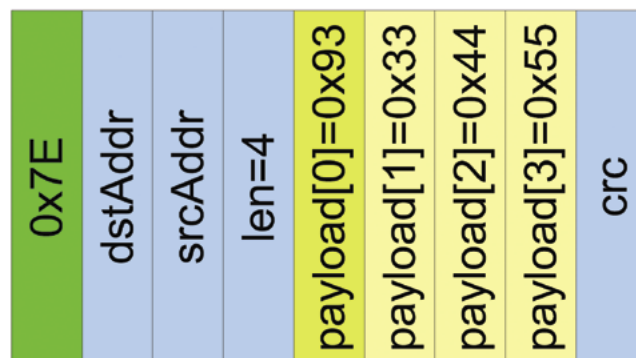
Jasności dla poszczególnych kanałów wpisujemy do „rejestrów jasności” pod adresami 101 do 110, natomiast czas przejścia pomiędzy jasnościami jest zależny od liczb wpisanych do „rejestrów opóźnienia” w przestrzeni adresów od 201 do 210. Rejestrom jasności odpowiada w programie głównym tablica *reg\_powerSet*, natomiast rejestry opóźnienia reprezentowane są tablicą *reg\_delay*. Każdy rejestr opóźnienia mówi ile (mniej-więcej) w milisekundach będzie trwała zmiana jasności kanału o 1. Wpisanie nowej (innej niż poprzednia) liczby do rejestrów jasności spowoduje stopniowe przejście do tej jasności po jednym



SPP\_ID\_CHAN\_VALUE | 0x03 – polecenie z maską bitową kanału nr 3.

Nowy poziom jasności

Rysunek 8.



SPP\_ID\_CHAN\_VALUE | 0x03 – polecenie z maską bitową pierwszego kanału, od którego chcemy rozpocząć ustawianie

Nowe poziomy jasności

Rysunek 9.



Tabela 2. Wykaz poleceń obsługiwanych przez sterownik taśm LED z oprogramowaniem dla protokołu SPPoB

Kategoria komendy/identyfikatora	Nazwa komendy/identyfikatora w payload[0]	Kod	ustawienie	Opis
Ogólne	SPP_ID_ALL_ON	0x04	1	Włącz wszystkie kanały w urządzeniu na 255 lub jasności zadane uprzednio poleceniem SPP_ID_POWER_DEFAULT.
	SPP_ID_ALL_OFF	0x05	1	Wyłącz wszystkie kanały w urządzeniu
	SPP_ID_ALL_ON_ALT	0x06	1	Jak SPP_ID_ALL_ON, tylko z efektem powolnej zmiany natężenia światła. O czasie przejścia dla każdego kanału decydują liczby zadane poleceniem SPP_ID_TRANSITION (domyślnie 10).
	SPP_ID_ALL_OFF_ALT	0x07	1	Wyłącz wszystkie kanały w urządzeniu używając efektu powolnego przejścia
	SPP_ID_MULTICHANNEL	0x08	≥4	Właściwa komenda zawarta w payload[3] i skierowana do wybranych kanałów, zadanych maskami w payload[1], payload[2]
Dla wybranych kanałów wyjściowych	SPP_ID_CHAN_ON	0x1n	1	Włącz kanał „n” (n=0x0...0x9)
	SPP_ID_CHAN_OFF	0x2n	1	Wyłącz kanał „n”
	SPP_ID_CHAN_TOGGLE	0x3n	1	Zmień stan kanału „n”
	SPP_ID_CHAN_ON_ALT	0x4n	1	Włącz kanał „n” w sposób alternatywny
	SPP_ID_CHAN_OFF_ALT	0x5n	1	Wyłącz kanał „n” w sposób alternatywny
	SPP_ID_CHAN_TOGGLE_ALT	0x6n	1	Zmień stan kanału „n” w sposób alternatywny
	SPP_ID_CHAN_UP	0x7n	1	rozjaśnij światło w kanale „n” o 10
	SPP_ID_CHAN_DN	0x8n	1	ściemnij światło w kanale „n” o 10
	SPP_ID_CHAN_VALUE	0x9n	1 + ilość kanałów do ustawienia	Zadaj wartość dla wybranych kanałów, rozpoczynając od „n”; jasność wpisujemy do payload[1, 2, ...], a o ilości kanałów do ustawienia decyduje długość pakietu
	SPP_ID_CHAN_VALUE_ALT	0xA n		J.w., tylko używamy efektu powolnego przejścia
	SPP_ID_TRANSITION	0xB n		Działa jak 0x9n tylko przesyłamy wartość opóźnienia dla efektów powolnej zmiany natężenia światła
	SPP_ID_POWER_DEFAULT	0xC n		Działa jak 0x9n tylko przesyłamy poziomy jasności ustawiane po wydaniu poleceń włączających kanał, np. SPP_ID_CHAN_ON
Komunikacja żądanie/odpowiedź	SPP_ID_REQ/SPP_ID_RESP	0xE0/0xF0	1/11	Odczyt zadanych poziomów jasności
		0xE1/0xF1		Odczyt poziomu jasności ustawianego po wydaniu poleceń włączających kanał, np. SPP_ID_CHAN_ON
		0xE2/0xF2		Odczyt chwilowych poziomów jasności
		0xE3/0xF3		Odczyt wartości opóźnień używanych do generowania efektów powolnej zmiany natężenia światła

kroku wykonywanym z opóźnieniem zadany w odpowiednim rejestrze opóźnienia. Np. w rejestrze jasności wybranego kanału mamy liczbę 100, a wpisujemy 150. W rejestrze opóźnienia dla tego kanału mamy liczbę 10, co odpowiada opóźnieniu ok. 10 ms na krok. Jasność zmieni się o: 150 – 100 = 50 jednostek, a przejście do nowej wartości będzie trwało 50·10 ms = 500 ms. Czas jednego kroku jest zadany stałym opóźnieniem w pętli głównej i efektywny czas opóźnienia jest zależny także od wykonania innych fragmentów kodu, stąd wspomnianą „ilość

**Wykaz elementów**

**Rezystory:**

R20...R29: 100 Ω (SMD 0603)  
 R1, R2, R5, R14: 470 Ω (SMD 0603)  
 R3, R4, R6: 1 kΩ (SMD 0603)  
 R30...R39: 2,4 kΩ (SMD 0805)  
 R10: 2,7 kΩ (SMD 0603)  
 R11: 4,7 kΩ (SMD 0603)  
 R8: 8,2 kΩ (SMD 0603)  
 R7, R12, R13: 10 kΩ (SMD 0603)  
 R9: niemontowany (SMD 0603)  
 R17...R19: zależne od aplikacji (SMD 0603)  
 RP1, RP2: drabinka 4x10 kΩ (SMD 1206)

**Kondensatory:**

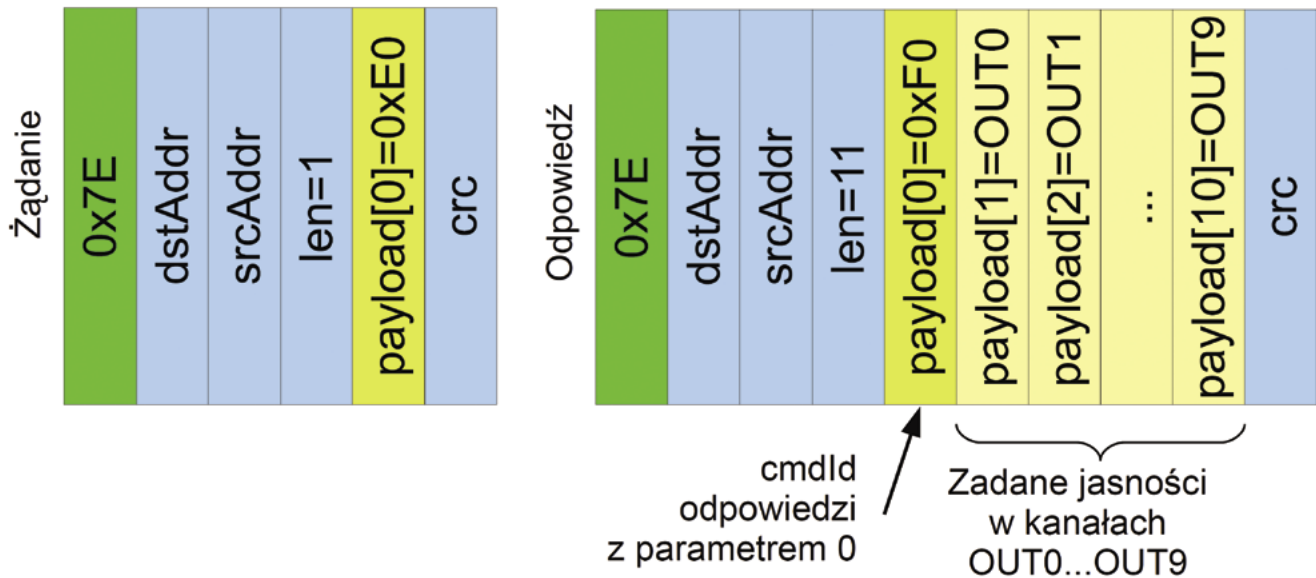
C7, C8: 10 pF (SMD 0603)  
 C13: 220 pF (SMD 0603)  
 C11: 22 nF (SMD 0603)  
 C2...C6: 100 nF (SMD 0603)  
 C14...C16: 1 μF (SMD 0805)  
 C1, C9, C17...C20: 10 μF (SMD 1206)  
 C10, C12: 100 μF (tantalowy, SMD rozm. D)

**Półprzewodniki:**

D1, D5: TVS 18 V jednokierunkowy, np. SMBJ18A  
 D3, D4: TVS 12 V jednokierunkowy, np. SMBJ12A lub dwukierunkowy  
 D2: Schottky, min. 1 A, SMD, np. SS14, SK26, STPS2L25U  
 IC1: STM32F103RBT6 (LQFP64)  
 IC2: L5972D (SO-8)  
 IC3: LM1117-3.3 (SOT-223)  
 IC4: MAX485 lub zamiennik (SO-8)  
 IC5...IC7: LM324 (SO-14)  
 LED1...LED3: diody LED (SMD 0805)  
 Q1...Q10 dla wersji P-MOS: AP4435GJ-HF-3TB lub zamiennik (TO-251)  
 Q1...Q10 dla wersji N-MOS: AP9962GJ-HF-3TB lub zamiennik (TO-251)

**Inne:**

CON1: goldpin 3p, 2,54 mm  
 CON2: goldpin 4p, 2,54 mm  
 CON3, CON5...CON14: złącze śrubowe 2p, raster 5,0 mm  
 CON4: złącze śrubowe 4p (lub 2x2p), raster 5,0 mm  
 F1: bezpiecznik polimerowy ok. 100 mA, maks. rozmiar 1812  
 F2: bezpiecznik polimerowy zależny od aplikacji, maks. rozmiar 1812  
 L1: koralik ferrytowy, np. 600 Ω @ 100 MHz (SMD 1206)  
 L2: dławik mocy, 33 μH, zalecany prąd pracy ≥0,5 A, maks. rozmiar 12x12 mm, SMD  
 SW1, SW2: Dipswitch 4p, (DIP-8)  
 VR1: warystor, napięcie pracy ok. 24 V DC (w podstawowej wersji nie montowany lub stosować z szybkim bezpiecznikiem przed sterownikiem)  
 X1: rezonator kwarcowy 8 MHz



Rysunek 10. Przykład sekwencji żądanie-odpowieź z parametrem 0 oznaczającym odczyt jasności

milisekund” należy przyjąć raczej jako wartość orientacyjną. Można oczywiście użyć bardziej precyzyjne odmierzenie czasu stosując inny mechanizm opóźnienia bazowego niż zastosowana, prosta funkcja delay\_ms o stałym opóźnieniu.

Do przeprowadzenia wstępnych testów działania sterownika w wersji z interfejsem Modbus można użyć dowolnego programu obsługującego protokół Modbus od strony urządzenia nadrzędnego. Przykładem funkcjonalnego, prostego w obsłudze i darmowego programu jest Modbus Tester dostępny pod adresem <http://goo.gl/TceCE5>. Na **rysunku 7** przedstawiono widok okna tego programu z objaśnieniami oraz ustawionymi parametrami do odczytu rejestrów jasności wraz z aktywnym okienkiem wpisu do rejestru o adresie 102.

### Interfejs zewnętrzny – protokół SPPoB

Wersja programu dla sterownika taśm LED z eksperymentalnym protokołem SPPoB ma rdzeń funkcjonalności bardzo zbliżony do wersji z protokołem Modbus, lecz ma nieco więcej opcji modyfikacji stanu wybranych kanałów. Po ogólny opis protokołu SPPoB odsyłam Czytelników do EP nr 10/2014, natomiast tutaj skoncentrujemy się na konkretnych przykładach.

Zestaw obsługiwanych komend przedstawiono w **tabeli 2**. Parametry transmisji są takie same jak dla interfejsu Modbus, tj. 19200 baud, 8 bitów danych, 1 bit stopu, brak bitu parzystości. Mechanizm działania przejść pomiędzy jasnościami (zaimplementowany w ledTask) działa także podobnie w obu wersjach.

Do testów warto skompilować źródła z makrodefinicją `SPP_IGNORE_CRC` ustawioną na „1” (plik `spp_cfg.h` w katalogu głównym projektu) i tak przygotowany plik binarny wgrać do pamięci Flash

mikrokontrolera. Dzięki temu będziemy mogli wysłać pakiety w formie makr z programu terminalowego `terminalbpp` ze strony <http://goo.gl/qnzBbj> i zamiast sumy kontrolnej na końcach pakietów wpisujemy dowolny bajt (w dalszych przykładach 0x00), a pakiet i tak zostanie przyjęty.

Zacznijmy od prostego ustawienia wartości w jednym kanale. Na rysunku 8 przedstawiono strukturę pakietu, w którym ustawiamy jasność kanału nr 3 na wartość 0x33 przy pomocy polecenia `SPP_ID_CHAN_VALUE` (definicja w pliku `spp_cmdid.h`). Założmy teraz, że pakiet wysyła urządzenie o adresie 0xA0, a sterownik pasków LED ma adres 0xB0. Wysyłany pakiet będzie zawierał ciąg liczb: [7E, B0, A0, 02, 93, 33, CRC]. Aby go wysłać, w programie terminalowym w polu wysyłania można wpisać ciąg znaków `$7E$B0$A0$02$93$33$00`. Zamiast sumy kontrolnej wysyłamy tutaj liczbę 0x00, natomiast znak „\$” w programie `terminalbpp` (lub Terminal by Br@y++) używany jest jako przedrostek liczby w formacie szesnastkowym.

Jeśli za pomocą tego samego polecenia `SPP_ID_CHAN_VALUE` zechcemy ustawić jasności w kanałach 3, 4 i 5 na odpowiednio 0x33, 0x44 i 0x55, to należy wysłać taki pakiet, jak na rysunku 9. Tak będą działały polecenia z tabeli 2 o kodach od 0x90 do 0xC0, dla których występuje parametr określający wartość nastawy.

Nieco prostsze będą polecenia jednobajtowe, służące do sterowania pojedynczych kanałów, o kodach 0x1n do 0x8n oraz ogólne: 0x04 do 0x07. Przykład pakietu włączającego domyślny poziom jasności w kanale 5 będzie następujący (adresy i zapis jak w poprzednim przykładzie): [7E, B0, A0, 01, 15, CRC].

Stosując komendę Multichannel możemy w jednym pakiecie wysłać jednakowe polecenie do wielu kanałów, np. wyłączyć wybrane kanały lub zadać im jednakowe poziomy

jasności. Składnia pakietu Multichannel została omówiona w pierwszej części artykułu o protokole SPPoB. Tutaj przedstawię jedynie prosty przykład. W przypadku omawianego sterownika, gdy chcemy przy pomocy polecenia `SPP_ID_CHAN_TOGGLE_ALT` (0x60) odwrócić stany w kanałach 9, 8, 7, 5, 2, 0 (ich maska bitowa to 0000 0011 1010 0101 czyli 0x03A5), to wyślemy np. następujący pakiet: [7E, B0, A0, 04, 08, 03, A5, 60, CRC].

I wreszcie możemy odczytać aktualne ustawienia dla sterownika wysyłając żądania z odpowiednimi parametrami. Przykład sekwencji żądanie-odpowieź z parametrem 0 oznaczającym odczyt zadanych jasności został przedstawiony na **rysunku 10**.

### Podsumowanie i dalsze pomysły

Z racji dostępności kodów źródłowych i oczywiście schematów sprzętu, sterownik pasków LED można bez trudu dostosować do realizacji własnego pomysłu: zarówno w dziedzinie nowych efektów, jak i interfejsu zewnętrznego. Ponadto płytka drukowana jest kompatybilna z wieloma mikrokontrolerami STM32F103, również takimi, które mają duże ilości wewnętrznych pamięci. Może to ułatwić zasycenie wewnątrz sterownika nawet całkiem rozbudowanej logiki działania. Wyprowadzony na zewnątrz interfejs RS-485 jest także bardzo uniwersalny i tani w implementacji. Dlatego nie powinno być trudności z wykonaniem przejśćówek z innymi interfejsów (USB, RS-232) lub wykonaniem zewnętrznych kontrolerów generujących polecenia Modbus lub SPPoB na podstawie danych odebranych np. z pilotów RC-5 czy modułów Bluetooth. Po modyfikacji oprogramowania urządzenie powinno nadawać się także do pracy jako generator „atmosfery” (ambient light) wokół monitora lub telewizora.

**Robert Brzoza-Woch**  
[robert.brzoza@gmail.com](mailto:robert.brzoza@gmail.com)