

Wyświetlacz LED z interfejsem 1-Wire

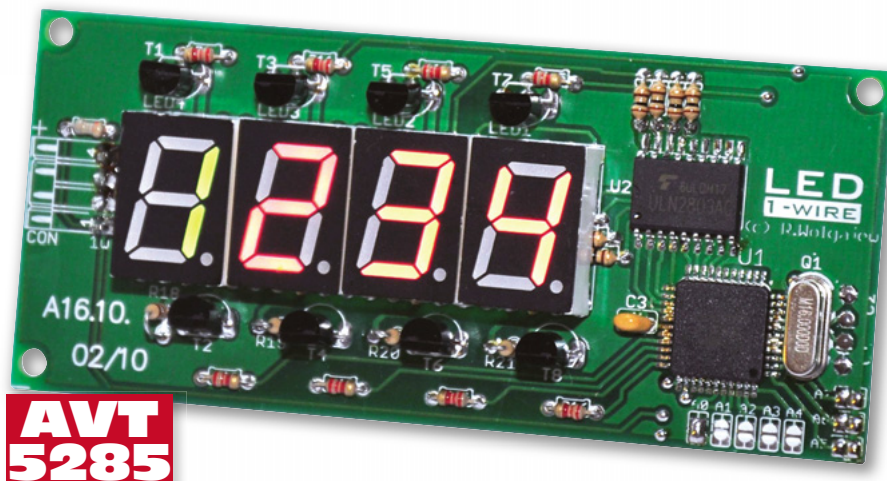


Niewątpliwą zaletą interfejsu 1-Wire jest niewielka liczba przewodów połączeniowych. W opisywanym projekcie zastosowano 1-Wire slave do dołączenia wyświetlacza LED umieszczonego w pewnej odległości od systemu sterującego jego pracą. Oprócz walorów użytkowych, projekt ma również walory dydaktyczne, ponieważ zaimplementowano w nim mechanizm wyszukiwania układów dołączonych do tej samej magistrali.

Rekomendacje: urządzenie może przydać się do wyświetlania np. temperatury czy czasu w miejscach odległych od systemu nadzorującego.

Chyba każdy elektroników spotkał się w swej praktyce z interfejsem 1-Wire. Ten ciekawy wynalazek opracowany przez firmę Dallas Semiconductor ma wiele unikalnych cech praktycznych czyniących z niego niespotykane medium transmisyjne. Niektórych Czytelników może dziwić nieco fakt, że ten interfejs jest stosowany jedynie w elementach produkowanych przez firmę Dallas-Maxim. Zapewne jest to wynikiem konieczności zakupu praw patentowych i niezbędnych licencji. Nic jednak nie stoi na przeszkodzie, aby użyć go w projekcie amatorskim. Nie mam tu na myśli konstrukcji układu typu Master, lecz zbudowanie „prawdziwego” układu Slave z interfejsem 1-Wire, obsługującego wszystkie rozkazy tejże magistrali, w tym mechanizm wyszukiwania adresów innych układów (*Search ROM*).

Jako przykład praktycznego rozwiązania urządzenia tego typu zostanie zaprezentowany moduł 4-cyfrowego, trójkolorowego wyświetlacza LED wyposażony we własny, unikalny numer identyfikacyjny (z możliwością zmiany). Rozwiązanie tego typu zostało wybrane nieprzypadkowo. W wielu systemach mikroprocesorowych zachodzi potrzeba użycia czytelnych i zarazem niedrogich, 7-segmentowych wyświetlaczy LED i jednoczesnej prezentacji wielu komunikatów.



AVT 5285

O ile sterowanie tego typu wyświetlaczami jest bardzo proste i może być zrealizowane przez najprostszego mikrokontroler, o tyle użycie większej liczby takich elementów następcza szereg problemów i stawia pod znakiem zapytania spełnienie niezbędnych warunków technicznych. Po pierwsze, nawet przy sterowaniu multipleksowym, użycie dużej liczby wyświetlaczy LED zajmie cenne porty mikrokontrolera oraz obciąży go programowo. Po drugie, może znacznie ograniczyć jasność diod LED, z których zbudowany jest każdy segment modułu.

Rozwiązaniem idealnym, jak się wydaje, jest zastosowanie odrębnych modułów wyświetlaczy wyposażonych we własny sterownik, adres i sterowanych ze wspólnej magistrali systemowej sterownika głównego. To właśnie zadanie realizuje przedstawiony moduł. Dzięki jego zastosowaniu tego budowa kilkupunktowego termometru staje łatwa i sprowadza się do zastosowania dosłownie kilku elementów: najprostszego, 8-wyprowadzeniowego mikrokontrolera obsługującego programową magistralę 1-wire, kilku termometrów scalonych DS1820 i przedstawionego modułu. A wszystko za pomocą pojedynczej magistrali 1-Wire.

Interfejs 1-Wire

W przypadku magistrali 1-wire, tak jak w przypadku większości interfejsów szeregowych, transmisja przebiega w konfiguracji Master-Slave. Układ nadrzędny (Master) steruje wyszukiwaniem i adresowaniem układów podrzędnych (Slave), przepływem danych (inicjuje wysyłanie i odbieranie da-

AVT-5285 w ofercie AVT:
AVT-5285A – płytka drukowana
AVT-5285B – płytka drukowana + elementy

Podstawowe informacje:

- Napięcie zasilania: 5 V
- Maksymalny prąd zasilania: 200 mA
- Liczba możliwych adresów sprzętowych: 256
- Liczba realizowanych poleceń: 4
- 3 kolory wyświetlania
- Ustawienie Fusebit'ów (zewnątrzny oscylator kwarcowy wysokiej częstotliwości):
 - CKSEL3:1: 111
 - CKSEL0: 1
 - CKOPT: 0
 - SUT1:0: 11
 - JTAGEN: 1
 - BODEN: 1
 - SPIEN: 0
 - OCDEN: 1
 - BOOTRST: 1

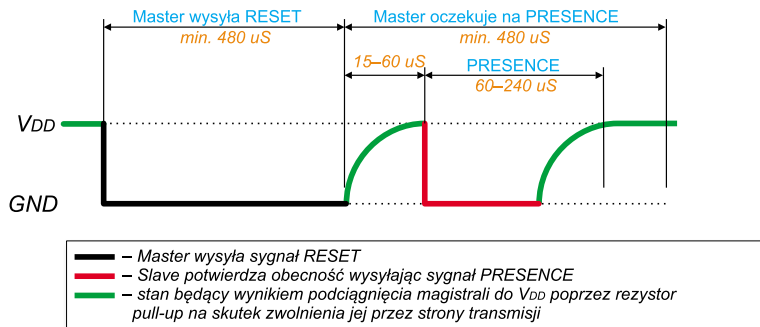
Dodatkowe materiały na CD/FTP:
<ftp://ep.com.pl>, user: 14039, pass: 4p80b5b5

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

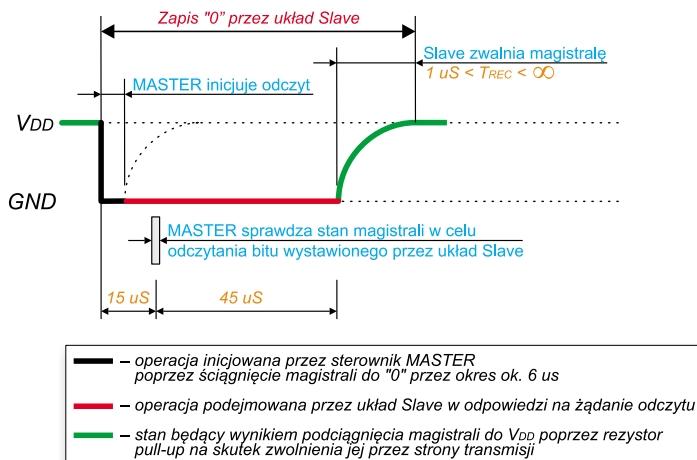
Projekty pokrewne na CD/FTP:
(wymienione artykuły są w całości dostępne na CD)

- AVT-5147 Moduł wyświetlacza LED sterowany magistralą I²C (EP 9/2008)
- AVT-934 8-cyfrowy wyświetlacz LED z interfejsem SPI (EP 6/2006)
- AVT-537 Moduł terminala z wyświetlaczem LED (EP 10/2003)
- AVT-859 Podwójny wyświetlacz siedmiosegmentowy sterowany I²C (EP 8/2000)

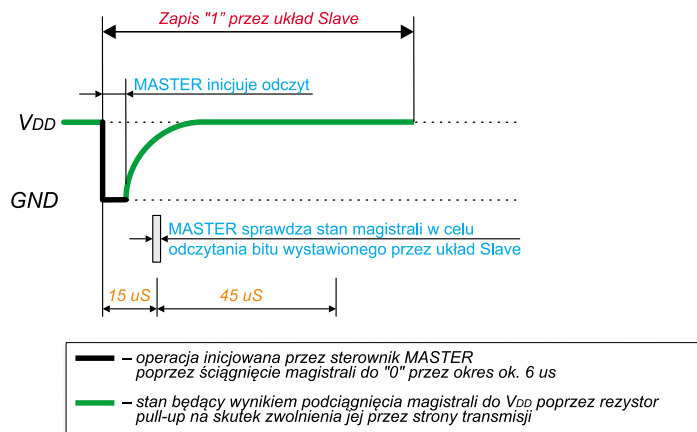
nych) oraz generuje sygnał zegarowy transmisji. Dane są przesyłane synchronicznie z prędkością do 16,3 kbps w trybie *standard* oraz do 115 kbps w trybie *overdrive*. Przesłanie każdego bitu informacji, niezależnie od kierunku transmisji, jest inicjowane wyłącznie przez układ Master za pomocą wygenerowania opadającego zbocza sygnału (ściągnięcie magistrali do logicznego „0” przez czas



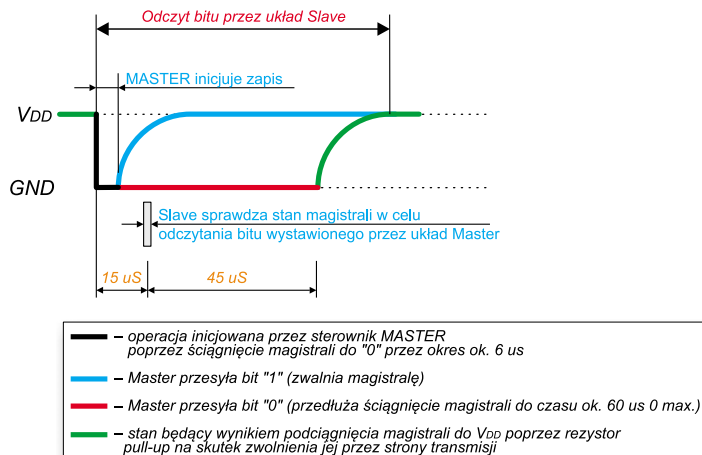
Rysunek 1. Procedura inicjalizacji magistrali 1-wire



Rysunek 2. Sekwencja sygnałów dla operacji zapisu logicznego „0” wykonywanej przez układ Slave, a inicjowanej przez układ Master.



Rysunek 3. Sekwencja sygnałów sterujących dla operacji zapisu logicznej „1” wykonywanej przez układ Slave, a inicjowanej przez układ Master.



Rysunek 4. Sekwencja sygnałów sterujących dla operacji odczytu stanu magistrali 1-wire wykonywanej przez układ Slave, a inicjowanej przez układ Master.

Wykaz elementów

Rezystory:

R1...R8: 100 Ω /0,125 W
 R9...R16: 2,2 k Ω /0,125 W
 R17: 3,3 k Ω /0,125 W⁽¹⁾
 R18...R21: 10 Ω /0,125 W⁽²⁾

Kondensatory:

C1, C2: 22 pF ceramiczny⁽³⁾
 C3: 100 nF ceramiczny

Półprzewodniki:

U1: ATmega16 w obudowie TQFP44
 U2: ULN2803 w obudowie SO18
 T1...T8: BC560
 LED4...LED1: wyświetlacz LED TOS5121

Inne:

CON – gniazdo 4-pin, kątowe, męskie (NSL25-4W)
 Q1 – rezonator kwarcowy 16 MHz (niski)

(1) W przypadku pracy kilku modułów 1-wire LED na jednej magistrali danych rezystor R17 zamontować należy wyłącznie na jednym z modułów.

(2) Rezystory R18...R21 są elementami opcjonalnymi niezbędnymi w przypadku zbyt małej różnicy jasności świecenia pomiędzy czerwonym a pomarańczowym kolorem modułów LED. W przypadku ich montażu należy zastosować pionowy sposób montażu.

(3) W przypadku problemów ze stabilną pracą oscylatora wbudowanego w mikrokontroler należy zrezygnować z montażu elementów C1 i C2 lub zastosować wartości zalecane przez producenta rezonatora kwarcowego.

około 6 μ s). Po wystąpieniu takiego zbocza, odbiorca adresu i polecenia, układ Slave podejmuje różne działania, których scenariusz zależy od oczekiwanego kierunku transmisji. Tego typu organizacja protokołu transmisji zapewnia prawidłową synchronizację przesyłanych danych bez potrzeby stosowania dodatkowych linii sterujących. Minimalny czas trwania pojedynczego bitu jest ściśle określony i wynosi 60 μ s + 1 μ s na tak zwany czas odtworzenia zasilania (*recovery time*). Wyznacza on maksymalną prędkość transmisji w trybie standard (1/61 μ s = 16,3 kbps). Każde z urządzeń dołączonych do magistrali musi mieć wyjście typu otwarty dren lub otwarty kolektor, a linia danych jest połączona jest do zasilania przez rezystor podciągający o wartości zależnej od długości połączenia. Ten rezystor w stanie beczynności powoduje utrzymywanie się poziomu wysokiego linii danych, co pozwala na zasilanie urządzeń podrzędnych (jeśli pracują one w trybie zasilania pasożytniczego). Sama magistrala nie ma ustalonego formatu danych, a sposób przesyłania informacji zależy

REKLAMA

od konfiguracji i właściwości układów podrzędnych. Przesyłane są słowa 1-bajtowe, w których jako pierwszy jest transmitowany bit najmniej znaczący. Dodatkową i jedną z najważniejszych cech urządzeń z interfejsem 1-Wire, odróżniającą je np. od urządzeń standardu I²C, jest unikatowy, ośmiobajtowy kod zapisany w pamięci ROM. Ten kod jest niepowtarzalny i przynależny tylko pojedynczemu układowi scalonemu (dla elementów produkowanych przez firmę Maxim/Dallas zapisywany jest na etapie produkcji). Najmniej znaczący bajt tego kodu zawiera numer rodziny układów (*Family code*). Kolejne 6 bajtów to unikatowy numer konkretnego egzemplarza (właściwy adresu układu). Najbardziej znaczący bajt zawiera sumę kontrolną CRC8 (*Cyclic Redundancy Check*). Ta suma jest wyliczana na podstawie poprzednich siedmiu bajtów i ustalana na etapie produkcji (służy do kontroli poprawności transmisji).

Protokół transmisji danych definiuje kilka, podstawowych stanów pracy magistrali:

- sygnał *Reset*, wysyłany przez układ Master, będący żądaniem zgłoszenia się układów Slave,
- sygnał *Presence*, wysyłany przez układy Slave, będący potwierdzeniem obecności tych układów na magistrali danych,
- zapis logicznej „1” i „0”,
- odczyt logicznej „1” i „0”.

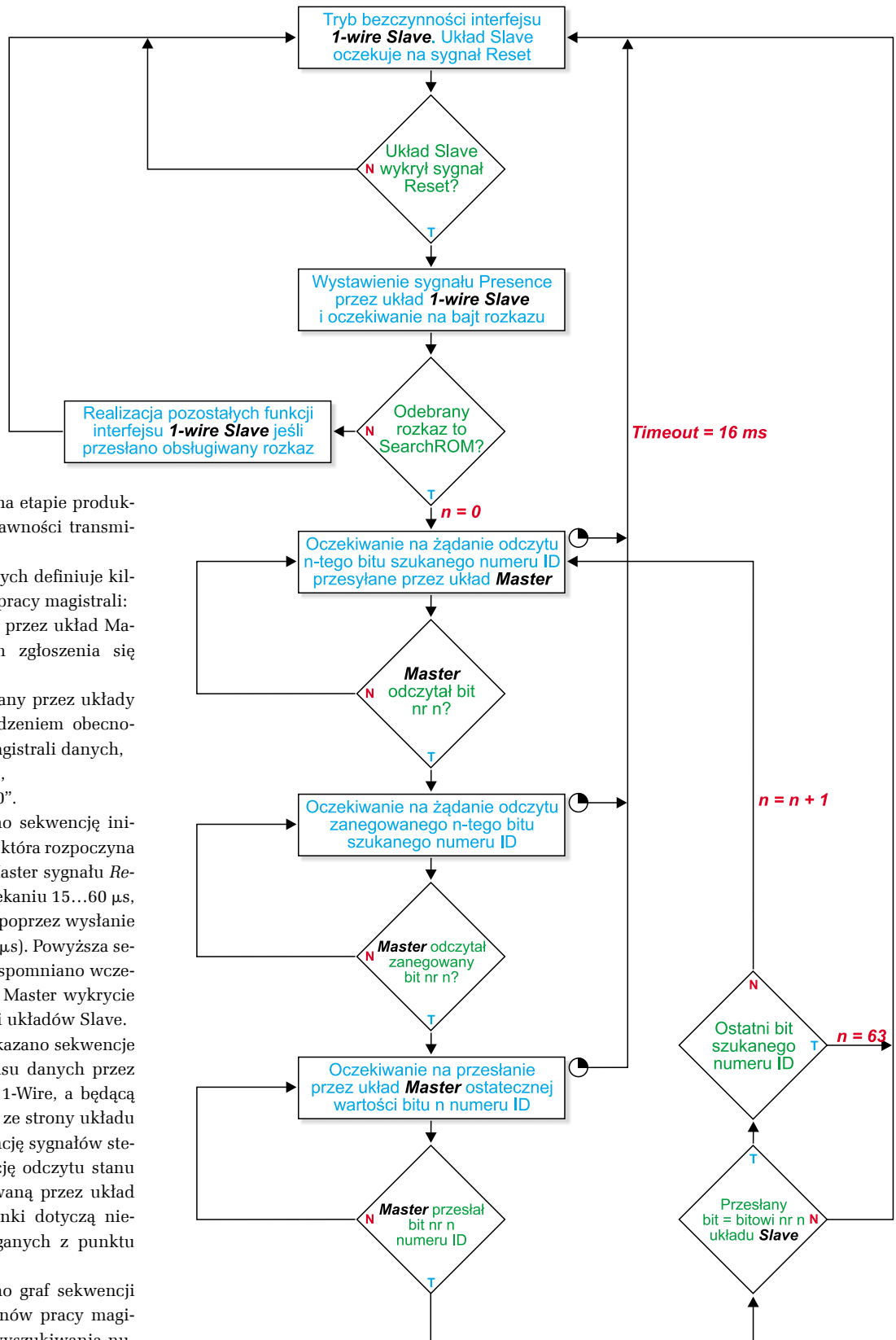
Na **rysunku 1** pokazano sekwencję inicjalizacji magistrali 1-Wire, która rozpoczyna od wysłania przez układ Master sygnału *Reset* (480...960 μs) i po odczekaniu 15...60 μs, odpowiedzią układu Slave poprzez wysłanie sygnału *Presence* (60...240 μs). Powyższa sekwencja inicjalizacji, jak wspomniano wcześniej, umożliwia układowi Master wykrycie podłączonych do magistrali układów Slave.

Na **rysunkach 2...4** pokazano sekwencje sygnałów dla operacji zapisu danych przez układ Slave na magistralę 1-Wire, a będącą wynikiem żądania odczytu ze strony układu Master, jak również sekwencję sygnałów sterujących ilustrujące operację odczytu stanu magistrali 1-Wire wykonywaną przez układ Slave. Przedstawione rysunki dotyczą niezbędnych działań postrzeganych z punktu widzenia układu Slave.

Na **rysunku 5** pokazano graf sekwencji sygnałów sterujących i stanów pracy magistrali 1-Wire dla operacji wyszukiwania numerów identyfikacyjnych (adresów) układów przyłączonych do 1-wire. Graf sporządzono

z punktu widzenia układu Slave. Operacja taka jest inicjowana poprzez wysłanie przez układ Master rozkazu 0xF0. Następnie każdy układ Slave przesyła na magistralę wartość pierwszego, najmniej znaczącego bitu swojego numeru identyfikacyjnego.

Mając na uwadze specyfikę interfejsu 1-wire, polegającą na tym, iż wszystkie układy są dołączone do tej samej magistrali danych, należy pamiętać, że na odebraną komendę przeszukiwania odpowiedzą dokładnie w tym samym czasie wszystkie układy Slave. Bity odbierane przez układ Master



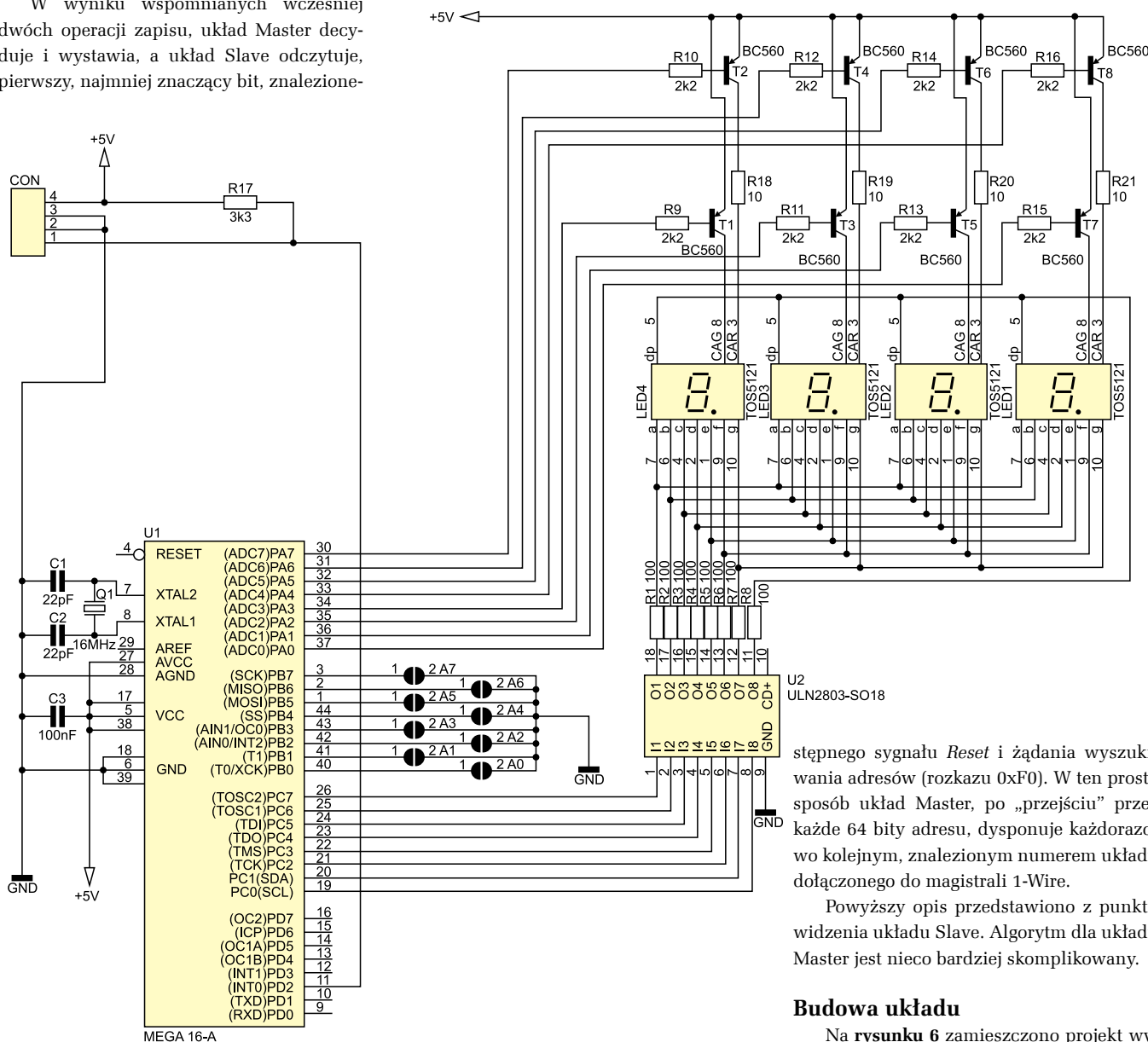
Rysunek 5. Sekwencja sygnałów sterujących obrazująca mechanizm znajdowania numerów ID układów Slave (z punktu widzenia tychże układów).

są iloczynem logicznym poziomów wyjść wszystkich układów (tzw. *wired and*). Kolejnym krokiem jest wystawianie przez układy Slave zanegowanego, pierwszego bitu swojego numeru identyfikacyjnego.

W wyniku wspomnianych wcześniej dwóch operacji zapisu, układ Master decyduje i wystawia, a układ Slave odczytuje, pierwszy, najmniej znaczący bit, znalezione-

go (w tej chwili przez układ Master) numeru identyfikacyjnego. Jeśli przesłana przez układ Master wartość (w tym trzecim kroku) pierwszego bitu numeru ID jest zgodna z rzeczywistą wartością pierwszego, najmniej

znaczącego bitu numeru ID wybranego układu Slave, układ ten kontynuuje proces aż do „przebrnięcia” przez wszystkie 64 bity numeru. Jeśli taka zgodność nie występuje, układ Slave pozostaje nieaktywny aż do na-



Rysunek 6. Schemat ideowy układu 1-wire LED.

stępnego sygnału *Reset* i żądania wyszukiwania adresów (rozkażu 0xF0). W ten prosty sposób układ Master, po „przejściu” przez każde 64 bity adresu, dysponuje każdorazowo kolejnym, znalezionym numerem układu dołączonego do magistrali 1-Wire.

Powyższy opis przedstawiono z punktu widzenia układu Slave. Algorytm dla układu Master jest nieco bardziej skomplikowany.

Budowa układu

Na rysunku 6 zamieszczono projekt wyświetlacza LED. Jest to nieskomplikowany

REKLAMA

Tabela 1. Wykaz poleceń 1-Wire obsługiwanych przez wyświetlacz

Rozkaz	Nazwa	Liczba bajtów towarzyszących	Opis rozkazu i znaczenie towarzyszących danych
0xCC	Skip ROM	5	Rozkaz ten przeznaczony jest do przesłania danych przeznaczonych do wyświetlenia dla wszystkich podłączonych do magistrali 1-wire układów Slave jako, że pomijany jest w tym przypadku docelowy numer ID. Układ Slave oczekuje w tym przypadku 5-ciu bajtów danych: Color Data1 Data2 Data3 Data4. Bajt oznaczony jako Color definiuje kolor każdego z segmentów LED przy czym bity 7...4 to składowe zielone koloru segmentów LED4...LED1 a bity 3...0 to odpowiednie składowe czerwone koloru wspomnianych segmentów. Bajty Data1...Data4 to wartości przeznaczone do wyświetlenia dla poszczególnych segmentów przy czym bit7 każdego z bajtów odpowiada za wyświetlenie kropki dziesiętnej. W ramach obsługiwanych znaków dostępne są cyfry 0..9 oraz znaki A...F (odpowiednie kody 0x00 do 0x0F). Dodatkowo przy użyciu kodu 0x10 możemy spowodować wygaszenie każdego z segmentów LED. Timeout dla tego rozkazu ustawiono na 5ms (o szczegółach tej funkcjonalności wspomniano wcześniej).
0x55	Match ROM	13	Rozkaz ten przeznaczony jest do przesłania danych przeznaczonych do wyświetlenia dla wybranego, specyfikowanego numerem ID, układu Slave. Układ Slave oczekuje w tym przypadku 13-ciu bajtów danych: 8-miu bajtów numeru ID oraz 5 bajtów specyfikujących kolor i wartości przeznaczone do wyświetlenia (jak dla rozkazu 0xCC). Timeout dla tego rozkazu ustawiono na 10 ms.
0x33	Read ROM	Nie dotyczy	Rozkaz ten służy do odczytania numeru Slave układu podłączonego do magistrali 1-wire, przy czym dotyczy przypadku, gdy na magistrali 1-wire pracuje wyłącznie jeden układ tego typu. W każdym, innym przypadku należy użyć rozkazu Search ROM (0xF0). W odpowiedzi na rozkaz Read ROM (0x33), układ Slave prześle swój 64-bitowy numer ID (oczywiście przesyłanie każdorazowo inicjowane jest przez układ Master). Timeout dla tego rozkazu ustawiono na 5ms.
0xF0	Search ROM	Nie dotyczy	Rozkaz ten inicjuje, opisany wcześniej mechanizm wyszukiwania adresów na magistrali 1-wire (Search ROM). Timeout dla tego rozkazu ustawiono na 16ms.

Listing 1. Procedura realizująca rozkaz Match ROM

```

lwreset      ,Sygnał Reset magistrali 1-wire
lwwrite &H55 ,Polecenie Match ROM (wybór konkretnego modułu)
, Następnie wysyłamy numer wybranego modułu 1-wire LED. Numer ten przechowuje 8-elementowa tablica Id()
, przy czym pierwsze 6 bajtów tego numeru to predefiniowane wartości (0x52 | 0x6F | 0x62 | 0x65 | 0x72 | 0x74),
, bajt 7. to sprzętowy adres modułu a bajt 8. to suma kontrolna obliczona przez program aplikacji z poprzednich 7 bajtów
danych.
lwwrite Id(1) , 8
, Wysyłamy bajt danych odpowiedzialny za kolor segmentów. W tym przypadku: wszystkie segmenty w kolorze czerwonym
lwwrite &B00001111
lwwrite &H01 ,Wartość dla pierwszego segmentu
lwwrite &H02 ,Wartość dla drugiego segmentu
lwwrite &H03 ,Wartość dla trzeciego segmentu
lwwrite &H04 ,Wartość dla czwartego segmentu
    
```

system mikroprocesorowy, w którym zastosowano tranzystory sterujące wspólnymi anodami wyświetlaczy LED, układ ULN2803, do którego dołączono katody wyświetlaczy LED oraz mikrokontroler Atmega16 pełniący rolę interfejsu magistrali 1-Wire i jednocześnie jednostki sterującej pracą wyświetlaczy. Wybór tego konkretnego typu mikrokontrolera z rodziny AVR wynikał wyłącznie z wymagań dotyczących liczby portów I/O oraz konieczności taktowania zewnętrznym rezonatorem kwarcowym.

W opisywanym układzie zastosowano dwukolorowe, 7-segmentowe wyświetlacze LED z niezależnymi wyprowadzeniami wspólnej anody dla każdej z barw (wyprowadzenia 8 i 3 elementu). Dodatkowym atutem zastosowanych elementów jest możliwość

jednoczesnej pracy obu diod świecących znajdujących się w każdym segmencie wyświetlacza, co w rezultacie daje nam możliwość uzyskania trzeciego koloru świecenia, w tym przypadku pomarańczowego. Zastosowano multipleksowe sterowanie każdym z modułów LED. Mikrokontroler, w odpowiedzi na dane otrzymane przy pomocy magistrali 1-wire, załącza wspólne anody kolejnych modułów LED „wystawiając” wcześniej odpowiednie wartości do portu PORTC, który to steruje połączonymi katodami wyświetlaczy. W ten prosty sposób, robiąc to odpowiednio szybko (w tym przypadku jest to częstotliwość ok. 100 Hz na każdy moduł LED), zapobiegamy zjawisku migotania wyświetlanych informacji, również w przypadku ich filmowania za pomocą kamery wideo.

Dodatkowo, na płycie sterownika, przewidziano 8 specjalnych punktów lutowniczych SMD przeznaczonych do sprzętowego ustawienia numeru ID.

Numer ten wyznaczamy według poniższego zapisu:

ID: 0x52 | 0x6F | 0x62 | 0x65 | 0x72 | 0x74 | HADR | CRC8,

gdzie:

HADR – to adres sprzętowy wynikający ze stanów odpowiednich wyprowadzeń portu PORTB mikrokontrolera,

CRC8 – to obliczana przez urządzenie suma kontrolna.

Do implementacji programowego interfejsu 1-Wire wykorzystano przerwanie zewnętrzne INT0 wyzwalane zboczem opadającym na dedykowanym wyprowadzeniu

Listing 2. Procedura realizująca rozkaz Skip ROM

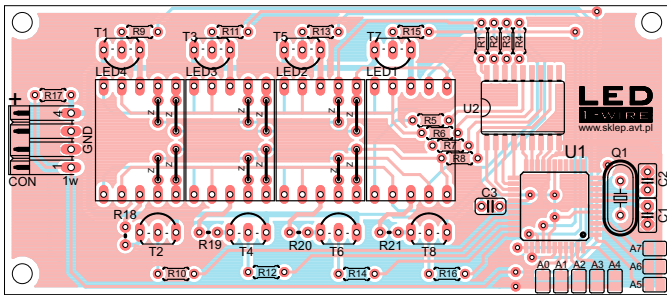
```

lwreset      ,Sygnał Reset magistrali 1-wire
lwwrite &HCC ,Polecenie Skip ROM (rozkaz dla wszystkich modułów 1-wire LED)
, Wysyłamy bajt danych odpowiedzialny za kolor segmentów. W tym przypadku: wszystkie segmenty w kolorze zielonym
lwwrite &B11110000
lwwrite &H0A ,Wartość dla pierwszego segmentu
lwwrite &H0B ,Wartość dla drugiego segmentu
lwwrite &H8C ,Wartość dla trzeciego segmentu (0x0C) plus kropka dziesiętna
lwwrite &H0D ,Wartość dla czwartego segmentu
    
```

Listing 3. Procedura realizującej rozkaz Read ROM

```

lwreset      ,Sygnał Reset magistrali 1-wire
lwwrite &H33 ,Polecenie Read ROM (rozkaz odczytania numeru ID modułu 1-wire LED - wyłącznie dla przypadku jednego modułu pracującego na magistrali)
Id(1) = lread(8) ,Odczytanie 8. bajtów danych numeru ID modułu 1-wire LED. Numer ten przechowa 8-elementowa tablica Id()
    
```



Rysunek 7. Schemat montażowy układu 1-wire LED.

mikrokontrolera jednocześnie dołączonym do linii danych. Co ciekawe, mimo iż interfejs 1-Wire jest bardzo wymagający pod względem zależności czasowych, cała procedura obsługi wspomnianego przerwania została napisana z użyciem Bascom Basic, który odkłada na stos zawartość większości rejestrów mikrokontrolera przedłużając w ten sposób obsługę każdego przerwania (przy ustawieniach standardowych) o 106 cykli zegarowych. Niemniej jednak, w tym konkretnym przypadku, rozwiązanie takie bez problemu wypełnia założenia protokołu 1-Wire.

Jak wspomniano, procedura obsługi przerwania INT0 odpowiada za odbiór i wykonanie wszystkich rozkazów interfejsu 1-wire ustawiając wartości odpowiednich zmiennych globalnych, a pozostawiając pętlę głównej jedynie realizację funkcji multipleksowania wyświetlaczy LED. Dodatkowo, wprowadzono mechanizm kontroli przebiegu transmisji, który po upływie pewnego czasu (tzw. timeout) i braku dalszej, oczekiwanej transmisji, powoduje przejście układu Slave do trybu bezczynności (oczekiwania na sygnał Reset) zerując wszystkie zmienne procedury obsługi przerwania INT0. W ten sposób zabezpiecza się układ Slave przez wejściem w tryb bezustannego oczekiwania np. w przypadku niepełnych, przerwanych czy nieprawidłowych ramek transmisji. Dla porządku należy dodać, iż program obsługi manipuluje wyłącznie rejestrem kierunku DDRD.2 portu PD2 mikrokontrolera realizując obsługę protokołu 1-wire. Przejdźmy, zatem do opisu dostępnych rozkazów sterujących obsługiwanych przez nasz sterownik. Ich listę z dokładnym opisem przedstawia zamieszczono w tabeli 1.

Na listingach 1...3 zamieszczono przykładowe procedury testowe odpowiedzialne za wykonanie rozkazów Skip ROM, Match ROM i Read ROM napisanych w języku Bascom Basic dla modułu 1-wire LED.

Montaż

Schemat montażowy układu pokazano na rysunku 7. Na początku należy wlutować elementów biernych (w tym rezystorów, które znajdują się pod wlutowanymi później modułami LED), następnie montujemy tranzystory, układy scalone (montaż powierzchniowy) a na końcu moduły wyświetlaczy LED oraz złącze. Pewnych kłopotów może nastęczyć montaż układów scalonych przeznaczonych do montażu powierzchniowego, zwłaszcza układu mikrokontrolera ATmega16 z uwagi na bardzo małe odległości pomiędzy poszczególnymi wyprowadzeniami. Montaż tego typu układu możemy wykonać na co najmniej dwa sposoby w zależności od sprzętu lutowniczego jakim dysponujemy. Sposób pierwszy to użycie specjalnej stacji lutowniczej (typu Hot Air) oraz odpowiednich, przeznaczonych do tego celu, topników. Sposób drugi (stosowany przeze mnie) to montaż przy użyciu typowej stacji lutowniczej, dobrej jakości cyny z odpowiednią ilością topnika oraz plecionki rozlutowniczej, która umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń układów. Należy przy tym uważać by nie uszkodzić termicznie układu. Poprawnie zmontowany układ nie wymaga jakiegokolwiek kalibracji i powinien działać bezpośrednio po włączeniu zasilania.

Robert Wołgajew
robert.wolgajew@ep.com.pl