

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za nieprawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

Mikroprocesorowy rejestrator „zdarzeń”

Wszelkie zdarzenia, które pojawiają się w losowych chwilach, a do tego bardzo nieregularnie są ze swej natury trudne do „wylapania”. Jeśli mają destrukcyjny wpływ, na przykład na działanie układu elektronicznego, to spędzają sen z powiek konstruktorom. Aby je wykryć i wyeliminować trzeba często odwoływać się niemal do czarów.

Rekomendacje:

kontroler „zdarzeń”, zgodnie z zamysłem autora może być cennym przyrządem pomocnym podczas uruchamiania urządzeń elektronicznych. Zakres zastosowań znacznie wzrośnie po dołączeniu interfejsu, np. generującego impuls cyfrowy wywołany dowolnym zdarzeniem fizycznym.



Projekt
162

PODSTAWOWE PARAMETRY

- Płytko o wymiarach 134x107 mm
- Zasilanie 5 V
- Liczba archiwizowanych sygnałów:
 - 9 (w trybie przenośnym),
 - nieograniczona (w trybie stacjonarnym)
- Zegar czasu rzeczywistego bez podtrzymania baterijnego

Podczas pracy pewnego układu, w pewnym jego fragmencie, zaczął się pojawiać z niewiadomych mi wtedy jeszcze przyczyn stan logiczny, który przeczył wszelkim zasadom logiki. Najpierw ustaliłem swoje podejrzenia, co do miejsca jego występowania. Z nadzieją spojrzełem na dostępny sprzęt, który mógł być mi pomocny. Niestety zwykły multimetr cyfrowy ze względu na swoją sporą bezwładność nie był w stanie wylapać zmian napięcia związanego z przejściem z jednego stanu w drugi i z powrotem. Analizator stanów logicznych? OK, zgodzę się, ale musiałbym sterczeć nad nim i wyczekiwać wystąpienia zmiany, którą zresztą mógłbym po prostu przegapić. No cóż, musiałem sobie jakoś pomóc budując najzwyklejszy przerzutnik, który zatrząskiwiał się w momencie wystąpienia niepożądanego stanu. Wiedziałem już gdzie. Nie wiedziałem jednak, kiedy. Pojawił się drugi problem. Ślęczenie nad układem nie należało do przyjemnych. Nie wiedziałem, czy będę czekał godzinę, czy godzin kilka? Czy problem leżał w przegrzewaniu się któregoś elementu, co wiązało się z jego dłuższą pracą, czy może występowały jakieś nieznanne mi zakłócenia? Swoją drogą, zbliżenie działającego telefonu komórkowego np. do ICL 7107 powoduje dosyć ciekawe i przypadkowe wskazania wyświetlacza. Postanowiłem wykonać układ na mikrokontrolerze i w najprostszy i znany wszystkim sposób sprzęgnąć go z układem czasu rzeczywistego, aby wiedzieć, o której godzinie wystąpił ten niekontrolowany stan.

I tak układ ewoluował, aż rozrósł się do obecnego projektu, który później wzbogaciłem o aplikację z nim współpracującą, pozwalającą archiwizować zgromadzone dane, co może okazać się przydatne.

Opis układu

Schemat układu kontrolera zdarzeń (rys. 1) można podzielić na kilka bloków. Podstawowym elementem jest mikrokontroler AVR ATmega8, takto-

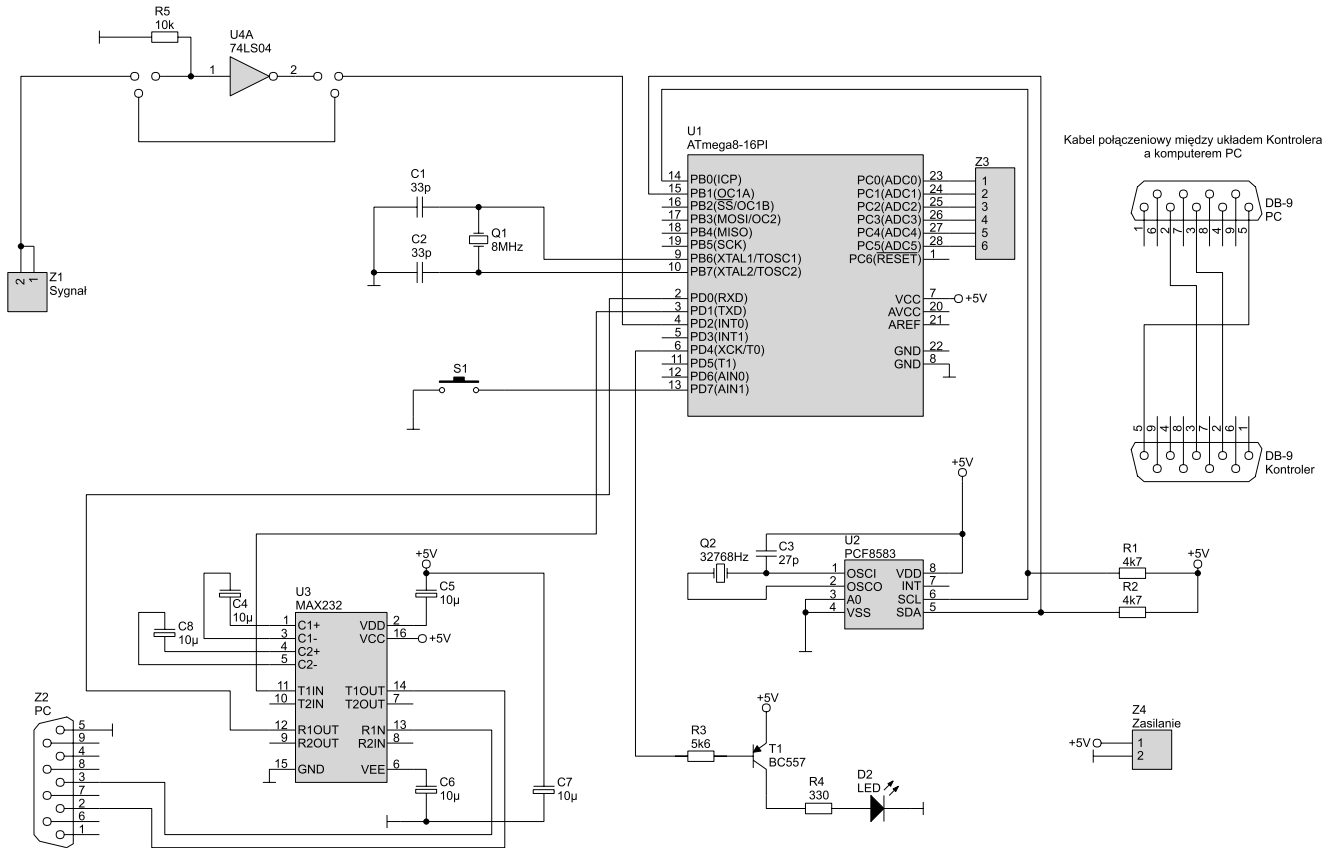
wany kwarcem 8 MHz. Jego program zajmuje ok. 3,5 kB pamięci Flash.

Drugim blokiem jest układ odpowiedzialny za kontrolę i podtrzymanie aktualnego czasu i daty. Oparty został na znanej wszystkim kostce PCF8583P, pracującej w podstawowej dla siebie konfiguracji, z wykorzystaniem magistrali I²C podciągniętej do napięcia zasilania poprzez rezystory R1 i R2 o wartościach 4,7 kΩ.

Ostatnim większym blokiem jest układ dopasowywania poziomu sygnałów interfejsu RS232C, zbudowany na kości MAX232 i kondensatorach uzupełniających.

Port PD.2 mikrokontrolera został skonfigurowany jako wejście wykrywające niskie stany logiczne. Aby nie ograniczać możliwości kontrolera, w jego obwodzie dorzuciłem bramkę NOT wziętą z kostki 74LS04. Z jej pomocą możemy wykrywać również zmiany na dodatni stan logiczny. Jeżeli sygnał badany, w stanie normalnym będzie logiczną „1” i interesować nas będzie jego przejście w stan „0”, to wówczas wejście podłączymy bezpośrednio do portu D.2 z pominięciem bramki NOT, odpowiednio konfigurując zworki. W przeciwnym przypadku, w którym interesuje nas przejście ze stanu „0” w stan „1”, wejście podłączymy właśnie z wykorzystaniem inwertera NOT. Rezystor R5 zapobiega utrzymywaniu się „pływającego potencjału” na niewykorzystywanym (ewentualnie) wejściu bramki U4A.

Podobnie wejścia pozostałych inwerterów zostały podpięte na sztywno do minusa zasilania (co jest widoczne tylko na płytce drukowanej). Niewielką różnicę w porównaniu ze schematem stanowi jeszcze złącze opisane jako PROG. Jest ono wyprowadzeniem programatora SPI, służącym do zaprogramowania mikrokontrolera. Podczas pisania i uruchamiania programu stanowiło ono duże ułatwienie pracy. W projekcie końcowym, wejście to nie zostało naniesione na schemat, gdyż mikrokontroler będzie już zaprogramowany docelowo. To czy zostanie ono zainstalowane na płytce drukowanej



Rys. 1. Schemat elektryczny kontrolera zdarzeń

pozostawiam decyzji samego użytkownika. Wzór płytki pokazany jest na rys. 2.

Uruchomienie

Przed rozpoczęciem konfiguracji układu należy połączyć go z komputerem poprzez port szeregowy. Uprzednio należy skonfigurować plik *Kontroler zdarzen.ini* oraz zmienić format daty systemowej wg schematu rrrr-mm-dd, co jest dosyć istotną sprawą, bez której nie ma mowy o poprawnym zadziałaniu układu. Teraz możemy uruchomić program i podłączyć napięcie zasilające. Tuż po uruchomieniu, zielona dioda znajdująca się na płytce powinna kilkakrotnie mignąć w odstępach ok. 1 s. Będzie to świadczyć o poprawnym zmontowaniu układu elektronicznego. Mając już gotowy do pracy układ, wybieramy jeden z dwóch dostępnych portów szeregowych oferowanych przez program i klikamy przycisk *Otwórz go* (rys. 3)

Aby sprawdzić czy rzeczywiście połączenie między komputerem a kontrolerem zostało poprawnie wykonane, możemy użyć opcji *Testuj połączenie*. O połączeniu lub też jego braku poinformuje nas stosowny komunikat. Jeżeli jesteśmy już pewni, że wszystko

jest w porządku, możemy przystąpić do dalszej konfiguracji kontrolera.

Opiszę teraz pokrótce znaczenie poszczególnych przycisków w kolejnych ramkach:

PORT RS232

Testuj połączenie – sprawdza poprawność połączenia między układem kontrolera a PC.

Otwórz port – otwiera wybrany przez nas port.

Zamknij port – zamyka wybrany przez nas port.

Skonfiguruj kontroler – przesyła do kontrolera wybraną przez nas konfigurację.

Pobierz ostatnią konfigurację – pobiera z bazy danych ostatnio utworzoną konfigurację kontrolera, którą każdorazowo przy zamykaniu programu możemy zapisać, co zostanie poprowadzone stosownym komunikatem wyboru. Warunkiem jest jednak zakończenie programu poprzez charakterystyczną ikonę wyjścia. Zamknięcie krzyżykiem nie da nam możliwości zapisania aktualnej konfiguracji.

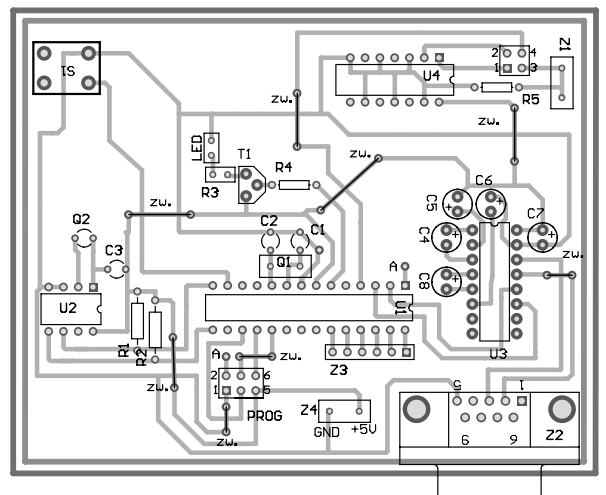
ODCZYTAJ/POBIERZ DANE

Analizuj dane – otwiera okno analizy danych zapisanych przez nas w bazie danych.

Pobierz dane – pobiera z mikrokontrolera zarchiwizowane dane i umieszcza w bazie. Aby pobrać dane należy po zakończeniu pracy kontrolera ponownie wyłączyć go i załączyć. Dotyczy to również jego ponownej konfiguracji.

TRYB PRACY

Praca przenośna – w trybie tym układ pracuje samodzielnie, archi-



Rys. 2. Schemat montażowy układu

wizując dane, które później zostaną zapisane w pamięci nieulotnej. Musimy pamiętać o tym, aby nie odłączać zasilania, bo spowoduje to utratę aktualnej konfiguracji. W trybie pracy przenośnej kontroler może jednorazowo zarchiwizować do 9 wykrytych sygnałów, przy czym liczba ta również może być konfigurowana.

Praca stacjonarna – w trybie tym układ pracuje na stałe podłączony do komputera. W tym przypadku liczba zarchiwizowanych informacji jest praktycznie nieograniczona, gdyż kontroler każdorazowo wysyła informacje do komputera, które automatycznie są zapisywane w bazie.

NASTAWY CZASOWE

Nastaw czas – konfiguruje czas i datę kontrolera (układ PCF8583P) na podstawie daty komputera.

Załącz – nastawia godzinę, o której kontroler zacznie analizować stan wejścia.

Wyłącz – nastawia godzinę, o której kontroler przestanie analizować stan wejścia.

Pracę można jednak rozpocząć wcześniej wykorzystując do tego celu przycisk S1

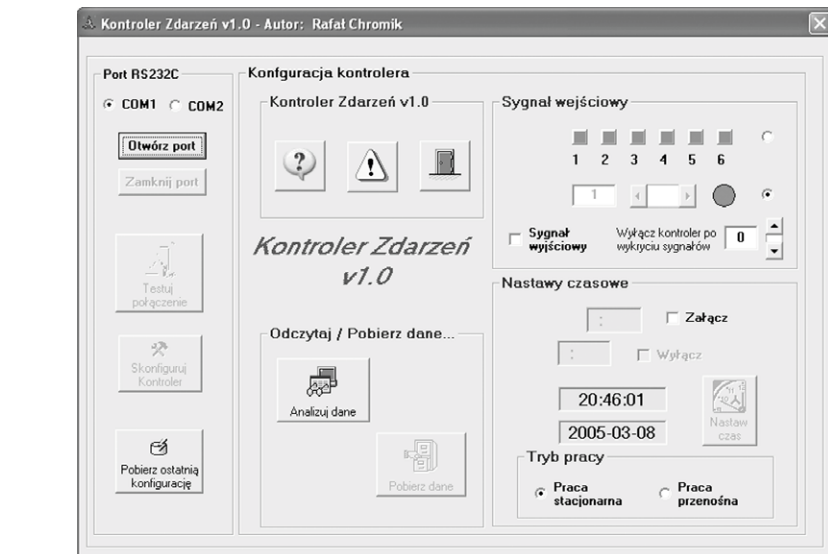
SYGNAŁ WEJŚCIOWY

Sygnal wyjściowy – zezwala na ustawienia na wyjściu takiego stanu, jaki ma przyjąć port C (złącze Z4) mikrokontrolera, po wykryciu pierwszego sygnału. Umożliwia to np. wysterowanie zewnętrznej elektroniki. Wartość wyjścia możemy ustalić na dwa sposoby – przypisując portowi liczbę z przedziału 1..63, lub po prostu, zaznaczając poszczególne jego wyprowadzenia od 1 do 6. Jeżeli sygnał będzie zanegowany, to na złączu Z4 będą panować logiczne zera.

Wyłącz kontroler po wykryciu sygnału – umożliwi zadeklarowanie liczby sygnałów, po których kontroler przestanie analizować sygnał. Jeżeli układ będzie miał zaprogramowany czas wyłączenia, a wcześniej osiągnie skonfigurowaną liczbę wykrytych zdarzeń, to wówczas wyłączy się.

Jako pierwszą czynność proponuję wykonać nastawę aktualnego czasu.

Po konfiguracji kontrolera, układ jest już w trybie pracy i może rozpocząć zbieranie danych. Oczywiście wystartować go trzeba za pomocą przycisku S1. Jeżeli pracuje w trybie przenośnym, to możemy pozostawić go samego. Po zgromadzeniu informacji, ponownie podłączamy komputer, restartujemy kontroler i wybieramy opcję – *Pobierz dane*. W tym momen-



Rys. 3. Okno aplikacji sterującej kontrolera zdarzeń

cie kontroler prześle do komputera zapisane dane, które trafią do bazy danych. Operacja trwa kilkanaście sekund, w zależności od liczby danych. Wysyłane są one co 1 sekundę. Będziemy mogli je już za chwilę obejrzeć wybierając opcję – *Analizuj Dane*.

Przyciskiem S1 umieszczonym na płycie możemy zatrzymać działanie kontrolera w wybranym przez nas momencie, w trakcie jego pracy. W czasie pracy, mogą wystąpić kilkunasto- lub kilkudziesięciosekundowe rozbieżności godziny załączenia i wyłączenia się kontrolera (przy wykorzystaniu opcji *Załącz/Wyłącz*).

Instalacja programu

Program „Kontroler zdarzeń” został ograniczony w wersji instalacyjnej do pliku wykonywalnego wraz z niezbędnymi bibliotekami. Po rozpakowaniu pliku tworzymy dowolny katalog w wybranym przez nas miejscu. Następnie kopiujemy tam wszystkie pliki. Teraz otwieramy *Kontroler zdarzen.ini* i wpisujemy ścieżkę dostępu dla naszej bazy danych *db1.mdb*, przy czym należy zwrócić uwagę na to, aby plik bazy danych znajdował się dokładnie w tym samym katalogu. Ważną kwestią jest również to, aby ścieżka dostępu podana w pliku *.ini* kończyła się znakiem \.

Ostatnią czynnością będzie przekopiowanie (nie przeniesienie) wszystkich plików, oprócz bazy danych i plików posiadających w swojej nazwie „Kontroler zdarzeń” do katalogu System32 bądź System, w zależności od wersji Windows. W ten sposób uzupełnimy brakujące biblioteki.

W tym momencie możemy uruchomić program.

Aby sprawdzić poprawność naszej „instalacji”, po uruchomieniu programu wybieramy opcję *Analizy danych* i klikamy na przycisk z okiem. Jeżeli nie pojawi się komunikat o błędzie, znaczyć to będzie, że operacja została przeprowadzona pomyślnie.

Baza danych posiada rozszerzenie **.mdb*, co oznacza, że da się ją otworzyć programem Microsoft Access, jednak do poprawnej pracy nie jest on wymagany. Jest to właśnie zaleta sterowników JET4.0, których użyłem do komunikacji z bazą danych. Dodatkową ich zaletą jest również to, że nie wymagają konfiguracji źródeł ODBC, co konieczne by było w przypadku wykorzystania np. MS SQL Server czy Sybase.

Rafał Chromik
almatea5@poczta.onet.pl

WYKAZ ELEMENTÓW

Rezystory

R1, R2: 4,7 kΩ
R3: 5,6 kΩ
R4: 330 Ω
R5: 10 kΩ

Kondensatory

C1, C2: 33 pF
C3: 27 pF
C4...C8: 10 μF/16 V

Półprzewodniki

U1: Atmega8
U2: PCH8583P
U3: MAX232
U4: 74LS04
T1: BC557
LED: 3 mm

Inne

Z1, Z4: ARK2
Z2: DB9 – do druku
Z3: goldpin
Q1: 8 MHz
Q2: 32768 Hz