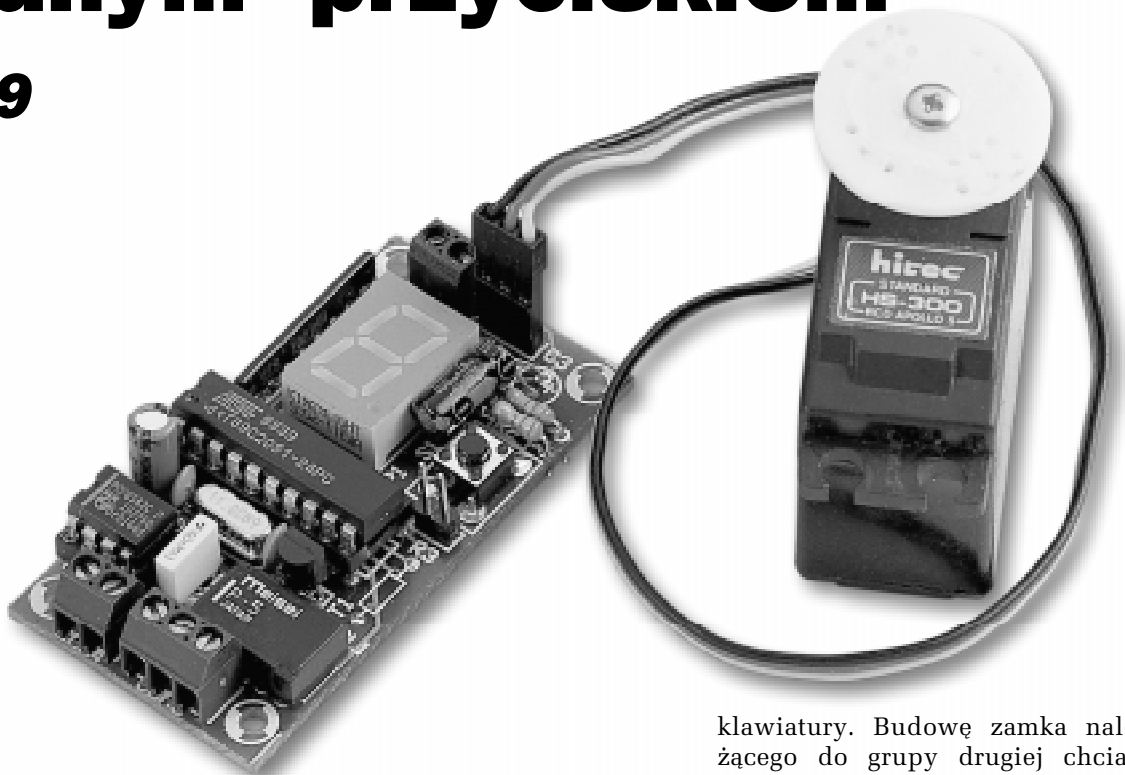


Zamek szyfrowy z jednym przyciskiem

AVT-869



Chciałbym zaproponować Czytelnikom budowę kolejnego zamka szyfrowego, czyli coś z grupy układów będących stałymi pozycjami „żelaznego” repertuaru pism przeznaczonych dla elektroników. W Elektronice Praktycznej opisano już wiele zamków i wyłączników szyfrowych, niemniej sądzę, że proponowane przeze mnie rozwiązanie jest na tyle oryginalne, że wzbudzi zainteresowanie Czytelników.

Zamki szyfrowe budowane są przez elektroników „od zawsze”. Pamiętam jeszcze projekty takich urządzeń realizowane wyłącznie na przekaźnikach, tańszych wówczas i łatwiej dostępnych niż tranzystory, nie mówiąc o układach scalonych.

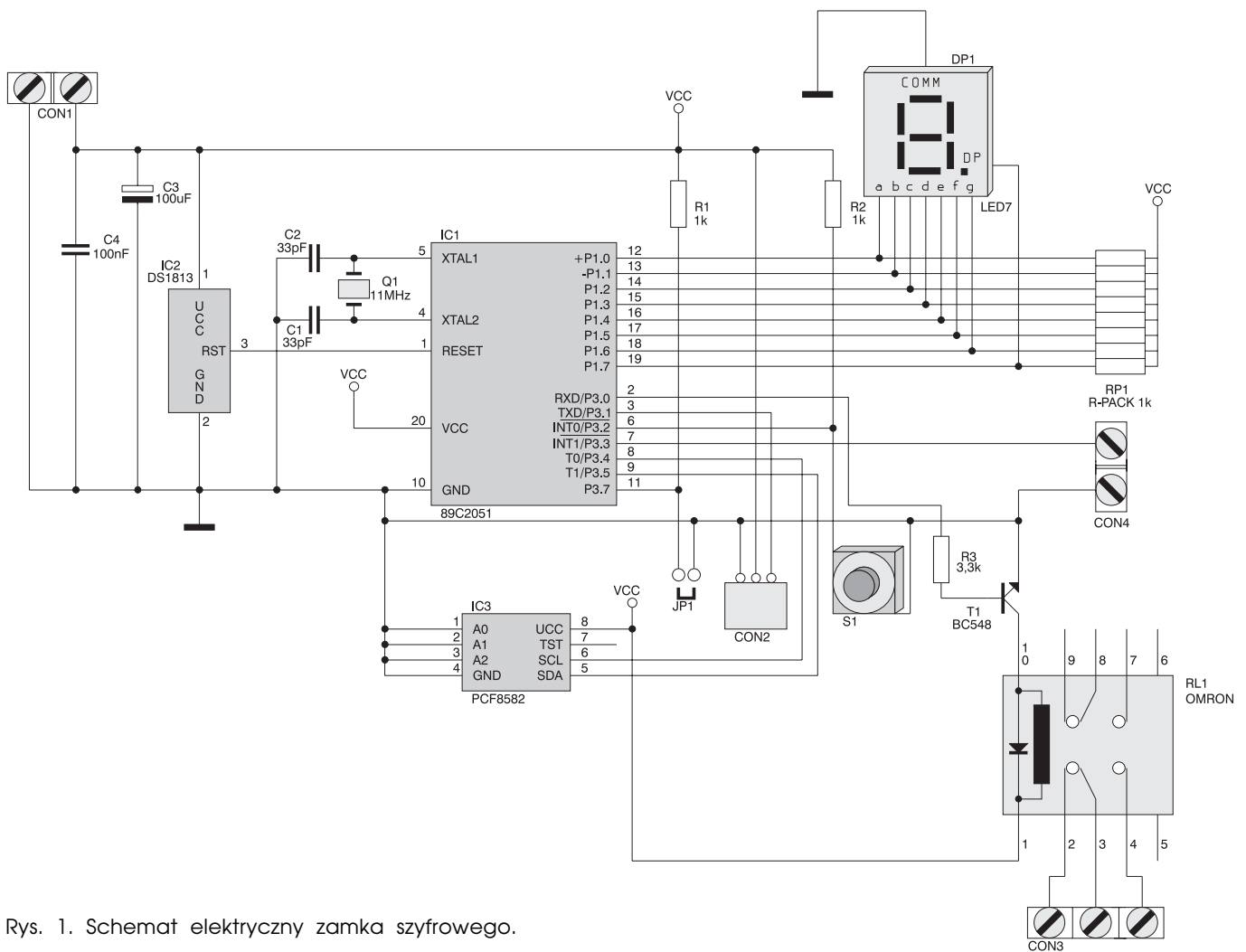
Moim zdaniem stosowanie typowej wieloprzyciskowej klawiatury w nowoczesnej konstrukcji zamka szyfrowego automatycznie dyskwalifikuje takie urządzenie i naraża konstruktora na posądzenie o pójście na łatwiznę, chyba że w założeniu miało być jedynie zabawką. Złamanie kodu takiego zamka jest zawsze dziecinnie łatwe, nawet bez analizowania stanu zużycia klawiszy.

Jakie jednak mamy rozwiązania alternatywne? Najogólniej mówiąc, zamki szyfrowe możemy podzielić na dwie kategorie: zamki otwierane za pomocą dołączenia do ich układu rozpoznawania elementu zewnętrznego o niepowtarzalnych cechach (pastylki DALLAS, karty magnetyczne, czytniki linii papilarnych itp.) oraz zamki, do których musimy wprowadzić kod ręcznie, najczęściej za pomocą

klawiatury. Budowę zamka należącego do grupy drugiej chciałbym zaproponować Czytelnikom.

Obawiam się, że w tym momencie zostałem posądzony o niekonsekwencję: z jednej strony krytykuję zamki z klawiaturami, a z drugiej zachęcam do wykonania takiej właśnie konstrukcji. Nie wspomniałem, że proponowany zamek będzie wprawdzie wyposażony w klawiaturę, ale z jednym tylko przyciskiem. Otwarcie zamka nastąpi po wprowadzeniu szyfru o praktycznie dowolnej (do 250) liczbie cyfr, ale nie będzie możliwe złamanie kodu ani za pomocą analizowania stanu zużycia klawiszy, ani przez proste podejrzenie osoby otwierającej zamek. Cała konstrukcja będzie znacznie bardziej zwarta i odporna na uszkodzenia, a i znalezienie odpowiedniej obudowy będzie o wiele łatwiejsze.

Zamki szyfrowe, jak ich nazwa wskazuje, służą najczęściej do zamykania drzwi wejściowych do pomieszczeń, kas pancernych itp. Do mechanicznego blokowania drzwi zwykle używane są tzw. rygle elektromagnetyczne, elementy niezbyt wygodne w użyciu, a przy tym mało odporne na uszko-



Rys. 1. Schemat elektryczny zamka szyfrowego.

dzenia mechaniczne. Drzwi zamknięte z wykorzystaniem typowego rygla elektromagnetycznego najczęściej możemy otworzyć bez znajomości jakiegokolwiek szyfru - wystarczy mocny kopniak! Najlepszym rozwiązaniem byłoby więc zastosowanie jako elementu blokującego otwarcie drzwi solidnej zasuwki napędzanej silnikiem elektrycznym. Dlatego też proponowany układ umożliwia bezpośrednie sterowanie serwomechanizmu - elementu idealnie nadającego się do przesuwania nawet bardzo solidnych rygla i zasuw. Do naszego układu możemy także dołączyć równolegle kilka serwomechanizmów pozwalających na symultaniczne poruszanie kilku rygla, co może być użyteczne przy konstruowaniu zapór szczególnie odpornych na próby sforsowania siłą.

Proponowany układ może także znaleźć zastosowanie jako szyfrowy wyłącznik dowolnych urządzeń elektrycznych, w tym central alarmowych. W tym celu został

wyposażony w przekaźnik o przełączanym styku.

Program sterujący pracą urządzenia został napisany, przetestowany i skompilowany za pomocą programu BASCOM 8951, opisywanego już na łamach Elektroniki Praktycznej. Jestem zagorzałym fanem rewelacyjnego pakietu BASCOM i dlatego opis działania układu ilustrowany będzie fragmentami kodu źródłowego programu procesora, napisanego w dialekcie BASIC-a.

Proponowany układ jest banalnie prosty i łatwy do wykonania nawet dla zupełnie początkujących konstruktorów.

Opis działania układu

Schemat elektryczny zamka szyfrowego pokazano na rys. 1. Sercem układu jest popularny, jakby stworzony na potrzeby hobbystów, procesor firmy ATMEL typu AT89C2051. Procesor ten posiada wiele zalet i jedną, dość poważną wadę: nie posiada wewnętrznej, nieulotnej pamięci danych typu

EEPROM. Bez takiej pamięci nasz układ działałby poprawnie, zapamiętywałyby wprowadzony kod, ale tylko do momentu zawsze mogącej się zdarzyć przerwy w zasilaniu. Po przywróceniu zasilania procesor podjąłby oczywiście normalną pracę, ale konieczne byłoby ponowne wprowadzenie kodu, a dostęp do strzeżonego pomieszczenia zostałby skutecznie zablokowany. Aby więc zabezpieczyć się przed taką ewentualnością, dodałem do układu zewnętrzną szeregową pamięć EEPROM typu PCF8582 - IC3. Jest to bardzo mała i tania pamięć, w której możemy zapisać tylko 255 bajtów danych. Jednak w naszym przypadku nawet taka pojemność pamięci nie zostanie najczęściej w pełni wykorzystana. Nie sądzę bowiem, aby ktoś chciał posługiwać się kodem dłuższym niż kilka, najwyżej kilkanaście cyfr, a w pamięci PCF8582 możemy w prosty sposób zapisać nawet liczbę 252-cyfrową (3 bajty pamięci zostały użyte do innych celów).

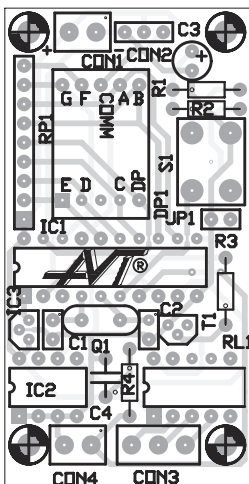
Analizę pracy układu, popartą fragmentami kodu źródłowego programu, rozpoczniemy w momencie narodzin naszego zamka szyfrowego, kiedy to zmontowany układ został dołączony do zasilania. W tym momencie pamięć EEPROM jest pusta i urządzenie nie byłoby w stanie normalnie pracować. Dlatego też podczas pierwszego uruchomienia układu, jak i podczas każdej zmiany kodu, musimy zewrzeć za pomocą jumpера JP1 pin 7 portu P3 do masy. Jest to dla procesora sygnałem, że ma umożliwić użytkownikowi wprowadzenie nowego kodu. Od tego momentu procesor pracuje w pętli programowej:

```

Do
For R = 1 To 10
'wyświetlanie kolejnych cyfr
'od 0 do 9
Set P3.7
P1 = Cyfra (r)
'kolejne cyfry zostały uprzednio
'zdefiniowane jako:
' Cyfra (1) = 63
' Cyfra (2) = 6
' Cyfra (3) = 91
' Cyfra (4) = 79
' Cyfra (5) = 102
' Cyfra (6) = 109
' Cyfra (7) = 125
' Cyfra (8) = 7
' Cyfra (9) = 127
' Cyfra (10) = 111

Enable Interrupts
Enable Int0
On Int0 Zapiszkod
'oczekiwanie na
'naciśnięcie przycisku

Wait 1
Disable Interrupts
    
```



Rys. 2. Rozmieszczenie elementów na płytce drukowanej.

```

Disable Int0
If Z = 252 Then
'jeżeli wprowadzono już
'252 cyfry kodu, to:
Call Write_eeeprom(255, Z)
'zapisz w komórce 255
'pamięci liczbę cyfr
P1 = 0 'wyłącz wyświetlacz
Exit Do 'wyjście z podprogramu
'wprowadzania kodu

Return
End If

If P3.7 = 1 Then
'jeżeli usunięto jumper
'JP1, to:
Call Write_eeeprom(255, Z)
'zapisz w komórce 255
'pamięci liczbę cyfr
P1 = 0 'wyłącz wyświetlacz
Exit Do 'wyjście z podprogramu
'wprowadzania kodu

Return
End If
Next R
Loop
    
```

Obserwując wyświetlacz, na którym cyklicznie ukazują się kolejne cyfry od 0 do 9, naciskamy przycisk w momentach, kiedy wyświetlona zostaje cyfra będąca kolejną pozycją kodu. Zapisywanie kolejnych cyfr kodu w pamięci EEPROM realizowane jest za pomocą krótkich podprogramów:

```

Zapiszkod:
Disable Interrupts
Disable Int0
Call Write_eeeprom(z, R)
Incr Z
Return

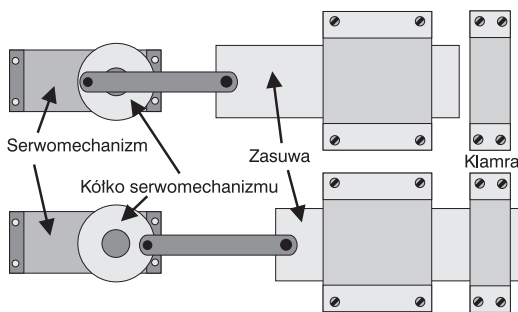
Sub Write_eeeprom(adres As Byte,
Value As Byte)
I2cstart 'inicjalizacja
'transmisji I2C
I2cwbyte Addressw
'podanie adresu bazowego
'pamięci EEPROM dla
'zapisu
I2cwbyte Adres
'podanie adresu
'wewnętrznej pamięci
I2cwbyte Value
'wartość do zapisania
I2cstop
'koniec transmisji I2C
Waitms 10
'przerwa 10 ms dla
'zapisania danych
'w EEPROM
End Sub
    
```

Wprowadzanie kodu trwa aż do momentu usunięcia jumpера JP1 lub do zapełnienia pamięci ponad 250 cyframi kodu. Jednak nie polecam nikomu stosowania kodu dłuższego niż 10 cyfr!

Po wprowadzeniu kodu i usunięciu jumpера układ rozpoczyna normalną pracę, która polega na oczekiwaniu na naciśnięcie przycisku S1. Procesor urozmaica sobie to oczekiwanie wysyłaniem krótkich impulsów na pin P1.7, co powoduje błyskanie punktu dziesiątego na wyświetlaczu i świadczy o pozostawaniu układu zamka w stanie czuwania. Naciśnięcie przycisku S1 spowoduje włączenie cyklicznego wyświetlania cyfr, a my obserwując wyświetlacz naciskamy przycisk w momentach ukazywania się na nim kolejnych cyfr ustawionego uprzednio kodu. Już po pierwszym naciśnięciu przycisku procesor odczytał z pamięci EEPROM liczbę cyfr występujących w kodzie, a teraz dokonuje sprawdzania poprawności każdej kolejnej cyfry. Jeżeli zostanie stwierdzone, że wybrane zostały właściwe cyfry, we właściwej kolejności i nie wybrano żadnej cyfry ponad te występujące w kodzie, to układ przystępuje do otwierania zamka.

Wybranie prawidłowego kodu powoduje ustawienie stanu wysokiego na wyjściu P3.0, a w konsekwencji przewodzenie tranzystora T1 i włączenie przekaźnika RL1. Jednocześnie procesor wysłała ciąg impulsów o czasie trwania 1 ms na wyjście P3.1, do którego za pośrednictwem złącza CON2 dołączone jest wejście serwowo-mechanizmu modelarskiego. Powoduje to ustawienie wału napędowego serwa w jednym ze skrajnych położen i odsunięcie rygla zamykającego drzwi.

Po upływie ok. 20 sekund na wyjściu P3.0 pojawia się stan niski, co powoduje rozwarzenie styków przekaźnika i wyłączenie dołączonego do niego urządzenia. Jednocześnie na wyjście P3.1 procesora wysłany zostaje ciąg impulsów prostokątnych o czasie trwania 2 ms, a w konsekwencji obrót wału napędowego serwowo-mechanizmu i ustawienie go w drugiej ze skrajnych pozycji. Rygiel zamka zostaje z powrotem zas-



Rys. 3. Sposób dołączenia serwomechanizmu do zasuwy drzwi.

nięty, a układ powraca do stanu oczekiwania na kolejne wybranie prawidłowego kodu.

Ważną rolę w układzie pełni IC2 - DS1813. Układ ten nie tylko generuje impuls resetu po włączeniu zasilania, ale także przez cały czas pracy procesora nadzoruje zasilające go napięcie. Jeżeli napięcie zasilania spadnie poniżej ok. 4,5V (w zależności od wersji układu DS1813 od 4,75 do 4,25V), to na wyjściu RST układu pojawia się wysoki poziom napięcia (stan 1), wymuszony za pomocą wewnętrznego rezystora o wartości typowo 5,5kΩ. Dlaczego zerowanie procesora i wstrzymywanie jego pracy podczas spadku napięcia jest takie ważne? Ano dlatego, że procesor zasilany nieodpowiednim dla niego napięciem zaczyna nieraz działać chaotycznie, wykonując czynności nie przewidziane przez programistę. Efektem takiej działalności może być (i często bywa) np. zamazywanie zawartości pamięci danych, co w przypadku naszego układu mogłoby spowodować zmianę ustawionego szyfru i niemożność otwarcia drzwi do zabezpieczonego pomieszczenia.

Montaż i uruchomienie

Na rys. 2 pokazano rozmieszczenie elementów na płytce obwodu drukowanego wykonanego na laminacie jednostronnym. Montaż układu rozpoczynamy od elementów o najmniejszych gabarytach, a kończymy na przekaźniku RL1 i kondensatorze

elektrolitycznym. Komentarza może wymagać jedynie wybór i montaż przycisku S1. W układzie modelowym, przeznaczonym do testowania układu w warunkach laboratoryjnych, jako S1 zastosowany został przycisk typu microswitch, przylutowany do płytki obwodu drukowanego. Jednak w wykonaniu układu użytkowego taki przycisk może okazać się zbyt delikatny i dobrze by było zastąpić go „czymś solidniejszym“, czyli przeznaczonym do długotrwałej pracy przyciskiem chwytowym dowolnego typu.

Gotowy układ powinien zostać zamontowany na drzwiach wejściowych w taki sposób, aby możliwa była obserwacja wyświetlacza. Idealnym rozwiązaniem wydaje się być zastosowanie jako wzornika gotowego elementu, tzw. „judasza“, w którym silnie rozpraszająca soczewka powinna być zastąpiona soczewką skupiającą lub zwykłą szybką szklaną. Sposób połączenia serwomechanizmu z rygłem zamka pokazuje rys. 3.

Układ powinien być zasilany napięciem stałym stabilizowanym o wartości 5VDC. Projektując zasilacz należy uwzględnić relatyw-

WYKAZ ELEMENTÓW

Rezystory

RP1: 1kΩ R-pack

R1, R2: 1kΩ

R3: 3,3kΩ

Kondensatory

C1, C2: 33pF

C3: 100μF/10V

C4: 100nF

Półprzewodniki

DP1: wyświetlacz siedmiosegmentowy LED wsp. katoda

IC1: AT89C2051 (zaprogramowany)

IC2: DS1813

IC3: PCF8582

T1: BC548

Różne

CON1: ARK2

CON2: 3x goldpin

CON: 3ARK3

JP1: jumper

Q1: rezonator kwarcowy 11,059MHz

RL1: przekaźnik OMRON 5V

S1: przycisk typu microswitch

nie duży (ok. 500mA) pobór prądu przez obciążony serwomechanizm i ewentualnie pomyśleć o stosowaniu zasilania alternatywnego, włączanego w momencie zaniku napięcia w sieci.

Zbigniew Raabe, AVT
zbigniew.raabe@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/pcb.html> oraz na płycie CD-EP10/2000 w katalogu PCB.

Kod źródłowy programu sterującego pracą mikrokontrolera znajduje się na płycie CD-EP10/2000.