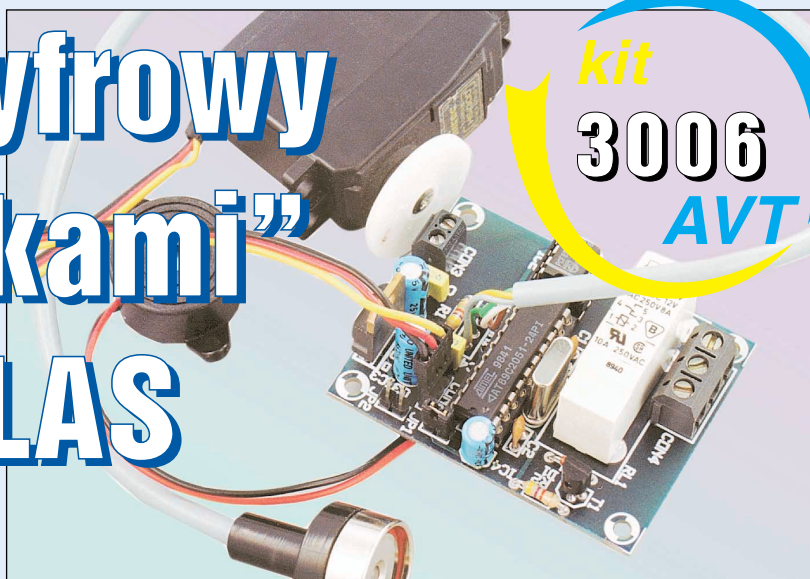




Zamek szyfrowy z "pastylkami" firmy DALLAS



kit
3006
AVT

Proponowany układ jest kolejnym urządzeniem, którego zadaniem jest zabezpieczenie naszego mienia przed osobami, które po wtargnięciu na nasz teren chciałyby tanim kosztem powiększyć stan swojego posiadania. Jest to po prostu zamek elektroniczny, ale o bardzo nietypowej konstrukcji, a przy tym nie dający złodziejom najmniejszej szansy na otwarcie go "sposobem". Po zbudowaniu tej konstrukcji długo starałem się opracować "patent" na otwarcie zamka bez posiadania właściwego klucza (oczywiście bez stosowania łomu czy acetylenowego palnika). Niestety, a właściwie na szczęście, nie znalazłem takiego sposobu i sądzę nie bez podstaw, że jest to po prostu niemożliwe. Niemniej, będę dożgonnie wdzięczny temu Czytelnikowi, który opracuje sposób otwarcia naszego zamka bez stosowania metod niszczących i podzieli się z nami swoim know how.

Układ, który dla Was zaprojektowałem może spełniać kilka zadań:

Podatkową jego funkcją jest praca jako praktycznie niemożliwy do sforsowania zamek do drzwi wejściowych, kasy pancerniej lub dowolnego innego pomieszczenia. **Jego dodatkową, bardzo atrakcyjną cechą jest wbudowany (a właściwie "wprogramowany" w procesor) układ sterowania serwo-mechanizmem, który może przesuwac nawet bardzo ciężkie rygle i zasuw.**

Proponowany układ można także zastosować jako szyfrowy włącznik – wyłącznik dowolnego urządzenia elektrycznego, chroniący je przed dostępem niepowołanych osób.

Trzecim możliwym zastosowaniem proponowanego układu jest użycie go jako immobilizera, czyli układu blokującego zapłon w samochodzie i tym samym znacznie utrudniający kradzież auta.

Nasz zamek szyfrowy może także zostać użyty do sterowania pracą central alarmowych i to zarówno przeznaczonych do ochro-

ny nieruchomych obiektów, jak i do strzeżenia samochodów.

Wykonanie układu z zaprogramowanym "fabrycznie" procesorem mogą polecić nawet mało doświadczonym Czytelnikom, którzy nie powinni natrafić na jakiegokolwiek trudności podczas jego budowy. Natomiast samodzielne zaprogramowanie procesora wymagać już będzie pewnej wiedzy, którą można było zdobyć w ramach kursu BASCOM College.

Jak to działa?

Zanim przejdziemy do szczegółowej analizy schematu zamka szyfrowego i sterującego jego pracą programu, powiedzmy sobie parę słów na temat zasady jego działania i kluczach stosowanych do jego otwierania. Z pozoru nie mam Wam do zaproponowania niczego nowego: zamek, jak zamek, do jego otwierania potrzebny jest klucz lub kilka pasujących kluczy. Tylko że kluczem do naszego zamka jest ... układ lub kilka układów scalonych i to o zaledwie dwóch wprowadzeniach!

Kluczami do proponowanego układu zamka szyfrowego są tzw. tabletki Touch Memory firmy DALLAS. Z wyglądu przypominają one zwykle baterijki od zegarków o nieco powiększonych wymiarach, ale w ich wnętrzu kryje się pamięć ROM z zapisanym w niej ośmiobitowym unikalnym numerem seryjnym i układ umożliwiający odczytanie zawartości pamięci. W przeciwieństwie do nawet najbardziej skomplikowanych kluczy mechanicznych na świecie nie ma dwóch identycznych tabletek DS1990!

Kolejną ciekawą cechą układów DS1990 jest sposób ich zasilania i transmitowania danych wspólny dla całej rodziny 1WIRE produkowanej przez DALLAS Semiconductors. Zarówno do zasilania układu, jak i do transmisji danych wykorzystywany jest tylko jeden przewód + masa. Układ zasilany jest me-

todą Parasite Supply (zasilanie pasożytnicze), wykorzystującą do ładowania kondensatora zawartego w układzie przerwy w transmisji danych. Pobór prądu przez układy 1WIRE jest tak mały, a transmisja danych przeprowadzana jest z taką szybkością, że taki sposób zasilania okazuje się całkowicie wystarczający.

Schemat elektryczny układu zamka szyfrowego został pokazany na rysunku 1. Łatwo zauważyć, że sercem układu i elementem spełniającym wszystkie jego ważniejsze funkcje jest procesor typu AT89C2051. AT89C2051 został już na łamach EdW omówiony bardzo szczegółowo, i wiemy, że procesor ten posiada rozliczne zalety i jedną, dość poważną wadę: nie posiada wewnętrznej nieulotnej pamięci danych typu EEPROM. Bez takiej pamięci nasz układ działałby poprawnie, zapamiętywał numery zarejestrowanych kluczy, ale tylko do momentu zawsze mogącej się zdarzyć przerwy w zasilaniu. Po przywróceniu zasilania procesor podjąłby oczywiście normalną pracę, ale konieczna byłaby ponowna rejestracja wszystkich kluczy, a dostęp do strzeżonego pomieszczenia zostałby skutecznie zablokowany. Aby więc zabezpieczyć się przed taką ewentualnością, dodałem do układu zewnętrzną szeregową pamięć EEPROM typu PCF8582. 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 potrzebował więcej niż 10 kluczy do jednego zamka, a do zarejestrowania takiej liczby pastylek DALLAS potrzeba zaledwie 80 bajtów pamięci! Pamięć typu PCF8582 została już także dokładnie opisana na łamach EdW, a w dalszej części artykułu opiszemy szczegółowo sposób jej adresowania i zapisywania danych.

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 też podczas każdej nowej rejestracji kluczy – pastylek DALLAS DS1990 musimy zewrzeć za pomocą jumpera JP2 pin 4 portu P3 do masy. Jest to dla procesora sygnałem, że ma przystąpić do rejestracji kluczy. Od tego mo-

mentu procesor pracuje w pętli programowej (tabela poniżej):

aż do momentu zarejestrowania wszystkich 10 kluczy. Uważni Czytelnicy z pewnością zauważą, że programista będący twórcą programu obsługującego nasz zamek próbuje jakby wymusić na Was konieczność posiadania i zarejestrowania zawsze 10 kluczy, co najczęściej jest liczbą znacznie przekraczającą rzeczywiste potrzeby. Tak jednak nie jest, możecie zarejestrować i używać każdą liczbę kluczy mniejszą lub równą 10. Po prostu, jeżeli np. chcecie zarejestrować tylko 3 klucze, to jeden z nich przykładamy do czytnika aż 7 razy! Może nie

jest to rozwiązanie zbyt eleganckie, ale za to znakomicie upraszczające konstrukcję układu.

Chciałbym jeszcze zwrócić Waszą uwagę na podprogram odczytywania numerów tabletek – kluczy i zapisywania ich w pamięci EEPROM (na następnej stronie). Jest to bowiem znakomity przykład siły pakietu BASCOM i używanego przez niego dialektu BASIC-a:

Chyba zgodzicie się ze mną, że stosowanie do programowania procesorów języków wysokiego poziomu nie jest pozbawione uroku! Wracajmy jednak do naszego układu.

Po zarejestrowaniu kluczy układ rozpoczyna normalną pracę, oczekując na przyłożenie do czytnika tabletki DS1990 i powzięcie decyzji czy jest to jeden z zarejestrowanych kluczy. Układ pracuje w pętli programowej, sygnalizując krótkimi błyskami diody LED czytnika, że jeszcze żyje i czuwa.

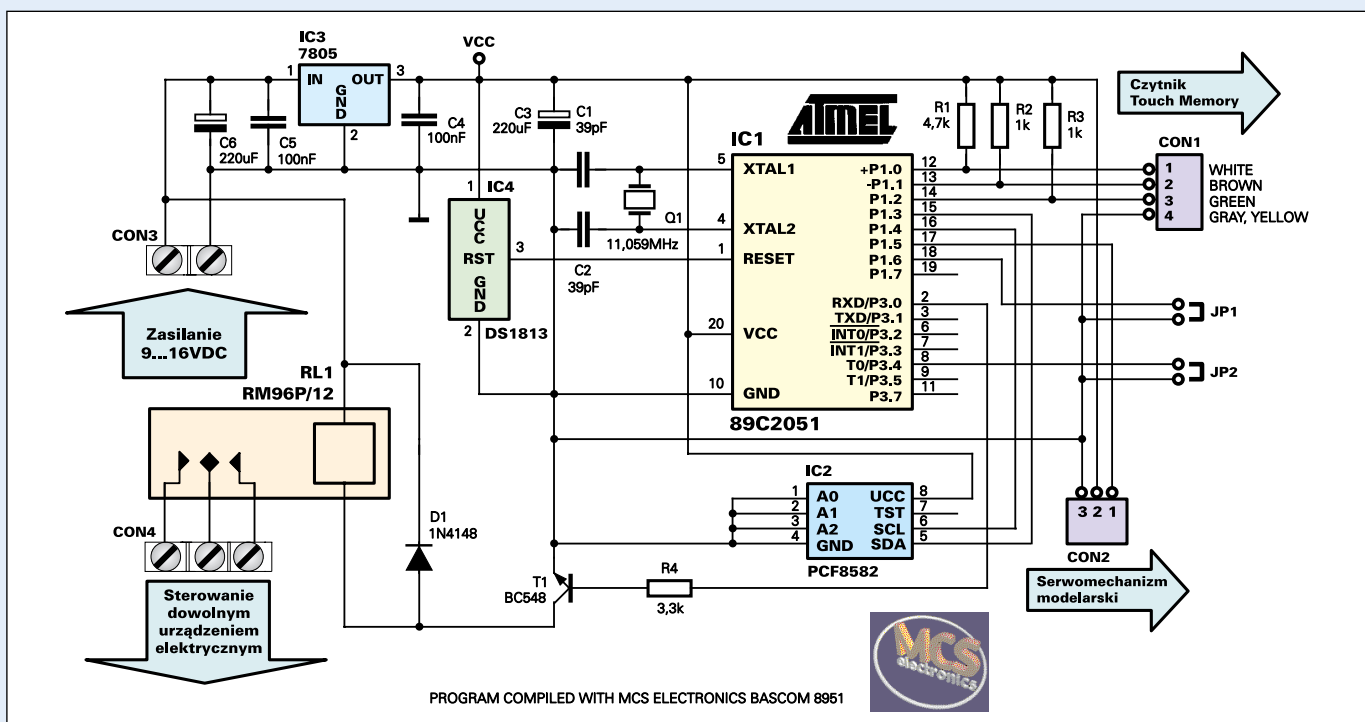
Po przyłożeniu tabletki do czytnika procesor odczytuje numer klucza i porównuje go kolejno ze wszystkimi numerami zapisanymi w pamięci EEPROM. Jeżeli wynik porównania jest negatywny, to długi błysk czerwonej diody i cztery sygnały akustyczne sygnalizują intruzowi, że "nie ze mną te numery, Brunner!". Natomiast reakcja układu na pozytywny wynik identyfikacji klucza zależy od położenia jumpera JP1.

1. JP1 rozwaroty. Po pozytywnym wyniku identyfikacji klucza na wyjściu P3.0 procesora pojawia się stan wysoki, co powoduje spolaryzowanie bazy tranzystora T1 i zwarcie styków przekaźnika RL1. Jednocześnie na wyjście P1.5 procesora wysłany zostaje ciąg impulsów prostokątnych o czasie trwania 1 ms. każdy. Wał napędowy serwomechanizmu dołączonego do złącza CON2

```

Rejestracja:
Licznik = 1      'ustawienie licznika adresu pamięci EEPROM
Do
1wreset        'sprawdzenie, czy do czytnika została przyłożona tabletki DS1990
If Err = 1 Then
Call Redlong   'jeżeli brak tabletki, to błyska czerwona dioda czytnika
Wait 1
End If
If Err = 0 Then
Call Beep     'po przyłożeniu tabletki sygnał akustyczny
Gosub Odczyt: 'skok do podprogramu odczytującego numer klucza i zapisującego go w
pamięci
Call Greenshort 'błysk zielonej diody potwierdza rejestrację klucza
Call Beep     'sygnał akustyczny potwierdza rejestrację klucza
Wait 1
End If
If Licznik > 80 Then 'sprawdzenie, ile kluczy zostało już zarejestrowanych, jeżeli 10 to:
Call Greenlong 'długi błysk zielonej diody potwierdza zarejestrowanie wszystkich kluczy
For Q = 1 To 5 'pięciokrotny sygnał akustyczny potwierdza j. w.
Call Beep
Next Q
Exit Do        'wyjście z pętli programowej
Return
End If
Loop
    
```

Rys. 1 Schemat ideowy

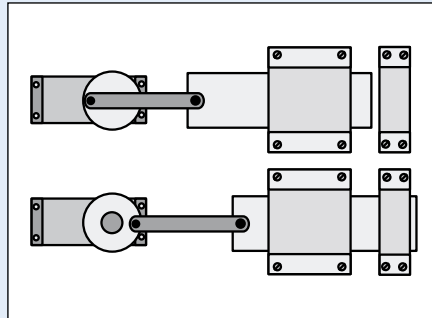


ustawia się w jednym ze skrajnych położeń, co powoduje odsunięcie rygla zamka zamykającego drzwi do strzeżonego obiektu.

Po upływie ok. 20 sekund na wyjściu P3.0 pojawia się stan niski, co powoduje rozwarcie styków przekaźnika i wyłączenie dołączonego do niego urządzenia. Jednocześnie na wyjściu P1.5 procesora wysłany zostaje ciąg impulsów prostokątnych o czasie trwania 2 ms. każdy, a w konsekwencji obrót wału napędowego serwomechanizmu i ustawienie go w drugiej ze skrajnych pozycji. Rygiel zamka zostaje z powrotem zasunięty, a układ powraca do stanu oczekiwania na dołączenie do niego klucza o prawidłowym numerze.

2. JP1 zwarty. Reakcja układu na pozytywny wynik identyfikacji klucza jest podobna do opisanej wyżej, z tą różnicą, że styki przekaźnika pozostają zwarte aż do momentu ponownego zidentyfikowania prawidłowego klucza. Podobnie, wał serwomechanizmu powróci do pierwotnego położenia dopiero po ponownym przyłożeniu do czytnika prawidłowego klucza.

pieczonych pomieszczeń. Sposób połączenia serwomechanizmu z rygłem zamka został pokazany na **rysunku 3**.

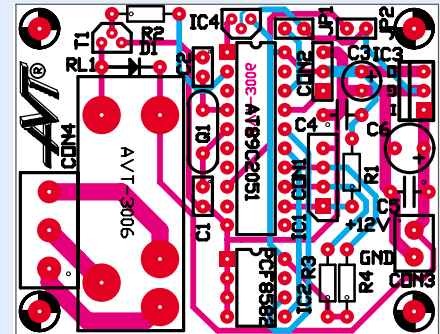


Rys. 3

Montaż i uruchomienie

Na **rysunku 4** została pokazana mozaika ścieżek płytki obwodu drukowanego wykonanego na laminacie dwustronnym z metalizacją oraz rozmieszczenie elemen-

Dalsze czynności, czyli rejestrowanie kluczy i posługiwanie się zamkiem zostały opisane wyżej.



Rys. 4

Uwaga:

W kicie dostarczane będą 2 tabletki DS1990. W razie potrzeby dowolną liczbę tych układów można zakupić oddzielnie.

Odczyt:

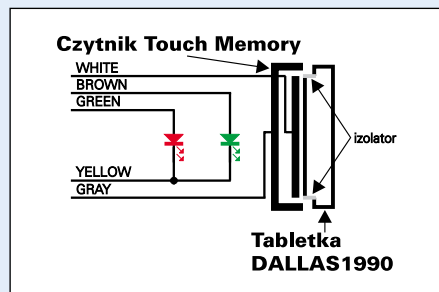
```
1wwrite &H33          'inicjalizacja układu DS1990
For I = 1 To 8
  Ar(i) = 1wread() :   'odczyt kolejnych bajtów ROM DS1990
  Next I
I = 0
For I = 1 To 8
  Call Write_eprom(Licznik , Ar(i)) 'zapis do pamięci EEPROM kolejnego bajtu numeru klucza
  Incr Licznik                'zwiększenie zawartości licznika adresu pamięci EEPROM
Next I
Return
```

```
Sub Write_eprom(adres As Byte , Value As Byte)
  I2cstart          'start transmisji I2C
  I2cwbyte Addressw 'podanie adresu pamięci EEPROM
  I2cwbyte Adres    'podanie adresu pod jaki mają zostać zapisane dane
  I2cwbyte Value    'przesłanie do pamięci bajtu informacji
  I2cstop           'zakończenie transmisji danych
  Waitms 10         'pauza 10 ms. dla umożliwienia zapisania danych w EEPROM-ie
End Sub
```

Budowa serwomechanizmów i sposób ich sterowania był już wielokrotnie opisywany w Elektronice dla Wszystkich. Warto natomiast dodać, że do naszego układu możemy dołączyć równolegle kilka serwomechanizmów, co pozwala na symultaniczne poruszanie kilku rygli, co może być użyteczne przy otwieraniu szczególnie dobrze zabez-

tów na płytce. Montaż wykonujemy w typowy sposób, rozpoczynając od wlotowania w płytkę elementów o najmniejszym gabarycie, a kończąc na przekaźniku i kondensatorach elektrolitycznych. Pod układy scalone obligatoryjnie stosujemy podstawki. Ze względu na szczególnie wysokie wymagania niezawodności, stawiane każdemu układowi zamka szyfrowego, powinny to być podstawki wysokiej jakości, najlepiej tzw. precyzyjne.

Po zmontowaniu układu i optycznym sprawdzeniu poprawności montażu wkładamy układy scalone w podstawki, zwieramy jumper JP2 i dołączamy do układu zasilanie o napięciu ok. 12VDC. Jeżeli w układzie zastosowany zostanie serwomechanizm, to projektując zasilacz, należy uwzględnić relatywnie duży prąd (ok. 500mA) pobierany impulsowo przez obciążone serwo.



Rys. 2

Zbigniew Raabe

Wykaz elementów

Kondensatory

- C1, C239pF
- C3, C6220μF/16V
- C4, C5100nF

Rezystory

- R14,7kΩ
- R2, R31kΩ
- R43,3kΩ

Półprzewodniki

- D11N4148
- IC1zaprogramowany procesor AT89C2051
- IC2PCF8582
- IC37805
- T1BC548
- IC4D8 1813

Pozostałe

- CON2goldpin x3
- CON3ARK2
- CON4ARK3
- JP1,JP22x goldpin + jumper
- Q111,059MHz
- Czytnik TOUCH MEMORY
- 2 tabletki DS1990 wraz z zawieszkami
- RL1przełącznik typu RM96P/12V
- Plezo z generatorem

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-3006