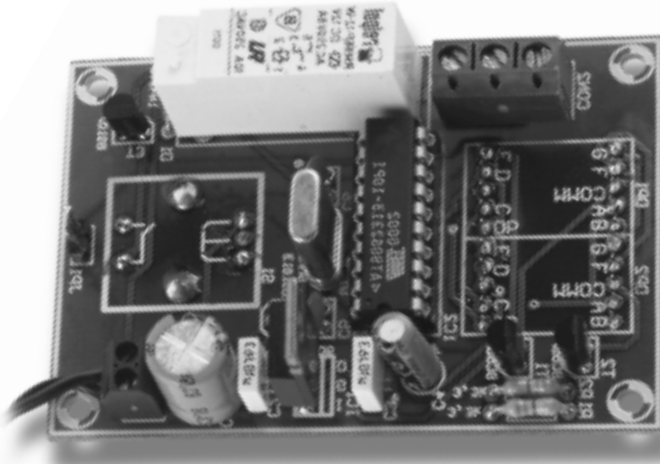


Zamek szyfrowy do sejfu

„Ucho od śledzia panie Kwinto”

AVT-5042

Kwinto podszedł wolnym krokiem do czającej się w kącie bankowego skarbcza bestii, którą był dla niego sejf znienawidzonego Kramera. Obecni, szeregowi pomocnicy od wykonywania czarnej roboty w rodzaju wysadzania dynamitem ściany skarbcza, wstrzymali oddech, kiedy mistrz położył ręce na lśniących pokrętlach zamka szyfrowego, arcydzieła dominującej w świecie, szwajcarskiej mechaniki precyzyjnej. Przez dłuższą chwilę było słycać jedynie cichutki terkot obracanych przekładni. Nagle dał się słyszeć niezbyt głośny trzask, potem drugi, trzeci... Kwinto pociągnął za uchwyt i drzwi sejfu zaczęły się powoli otwierać...



Piękna scena, piękne czasy! Czasy, w których królami przestępców nie byli zwykli bandyci, ale genialni kasiarze, otwierający każdy zamek bez używania siły i robienia komukolwiek krzywdy. Mechaniczne zamki do sejfów były prawdziwymi cudami ówczesnej techniki, obecnie zastąpionymi przez bezduszne zamki elektroniczne.

Spróbujemy jednak zbudować elektroniczny odpowiednik szyfrowego zamka do sejfu, który przy obecnym stanie elektroniki powinien być układem banalnie prostym. Tylko, że tym razem: ucho od śledzia, panie Kwinto, ucho od śledzia! Tego zamka nie otworzy nawet najgenialniejszy kasiarz, ani Pan ani nawet sam mistrz Szpicbródka! Nawet najlepszy słuch i największe „czucie” w palcach są bezradne wobec nowoczesnej elektroniki!

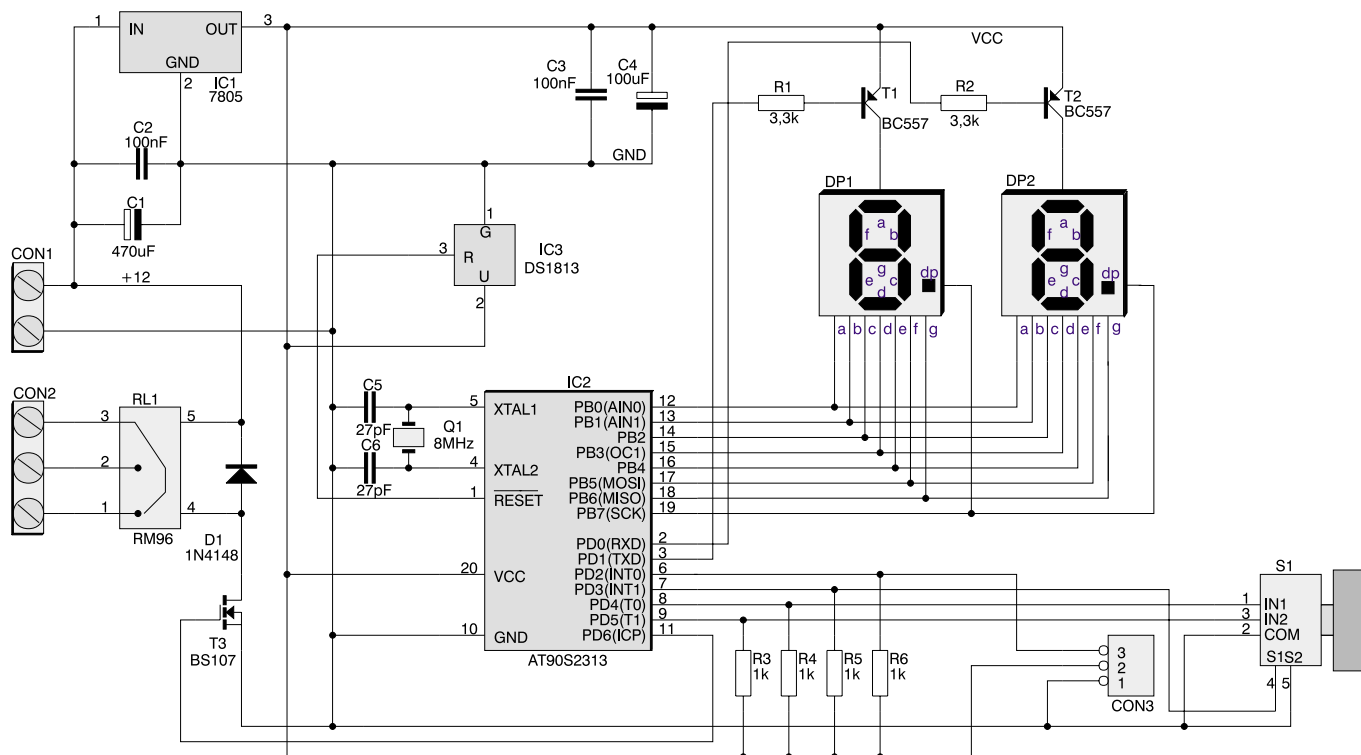
Prezentowany w artykule zamek może być zarówno niezwykle efektowną zabawką, jak i w pełni użytecznym urządzeniem zabezpieczającym dostęp, może nie do skarbcza bankowego, ale do domowego sejfu czy skrytki. Może być także łatwym w obsłudze i trudnym do otwarcia przez niepowołane osoby zamkiem do drzwi wejściowych domu lub innego pomieszczenia.

Kluczem do zamka jest dowolny ciąg liczb (**nie cyfr!**) dwucyfrowych. Dowolny, ponieważ ich maksymalna liczba wynosi 125, co znacznie przekracza możliwoś-

ci pamięciowe przeciętnego człowieka. Nawet przy wykorzystaniu kilku liczb odgadnięcie klucza jest praktycznie niemożliwe, a w każdym razie wymaga więcej szczęścia niż wylosowania „szóstki” w Toto Lotku.

Dokładne skopiowanie w sposób elektroniczny klasycznego zamka szyfrowego do sejfu nie miałyby chyba większego sensu. Prowadziłoby to do zastosowania kilku pokręteł, czyli do niepotrzebnego komplikowania konstrukcji oraz zwiększenia kosztów wykonania. Dlatego też w układzie zastosowałem tylko jedno pokrętko (impulsator obrotowy) oraz podwójny wyświetlacz siedmiosegmentowy LED, na którym jest prezentowane aktualne położenie pokrętkła zamka.

Nawet najlepszy zamek szyfrowy niewiele byłby wart bez układów wykonawczych. Układ naszego zamka został wyposażony w dwa wyjścia. W jednym zastosowano zwykły przekaźnik o dużej obciążalności styków, za pośrednictwem którego możemy sterować dowolnym urządzeniem elektrycznym nie pobierającym więcej niż 8A przy 220VAC. Natomiast do drugiego wyjścia możemy dołączyć typowy serwomechanizm modelarski. Po otwarciu zamka wał napędowy serwa ustawiany jest w jednym ze skrajnych położań, a po zamknięciu, w przeciwnym, po obróceniu wału o 180°. Serwomechanizm modelarski charakte-



Rys. 1. Schemat elektryczny zamka.

ryzuje się znacznym momentem obrotowym, co umożliwi przesunięcie nawet bardzo ciężkich zasuw, a nawet ich zespołów.

Układ zamka szyfrowego do sejf nie jest skomplikowany i jest możliwy do wykonania nawet przez zupełnie początkujących konstruktorów.

Opis działania układu

Schemat elektryczny układu zamka do sejf pokazano na rys. 1. Najważniejszym elementem, decydującym o jakości działania, jest zaprogramowany procesor typu AT90S2313, RISC-owy odpowiednik popularnego '2051. Procesor '2313 ma wbudowaną nieulotną pamięć danych EEPROM, co zwalnia z konieczności przechowywania w pamięci zewnętrznej liczb wchodzących w skład klucza. Drugim atutem procesora '2313, który przesądził o zastosowaniu go w układzie zamka, jest jego sprzętowy watchdog. Zgodnie ze swoim przeznaczeniem, układ zamka szyfrowego ma działać w stanie czuwania całe miesiące i lata. W ciągu tak długiego czasu zawieszenie procesora na skutek chociażby zakłóceń zewnętrznych jest więcej niż prawdopodobne. Przed takim właśnie przypadkiem strzeże nas sprzętowy watchdog.

Wszystkie wyjścia portu B procesora zostały wykorzystane do sterowania segmentami wyświetlaczy siedmiosegmentowych DP1 i DP2. Wspólne anody tych wyświetlaczy są zasilane za pomocą tranzystorów T1 i T2. Bazy tych tranzystorów sąysterowywane z wyjść 0 i 1 portu D.

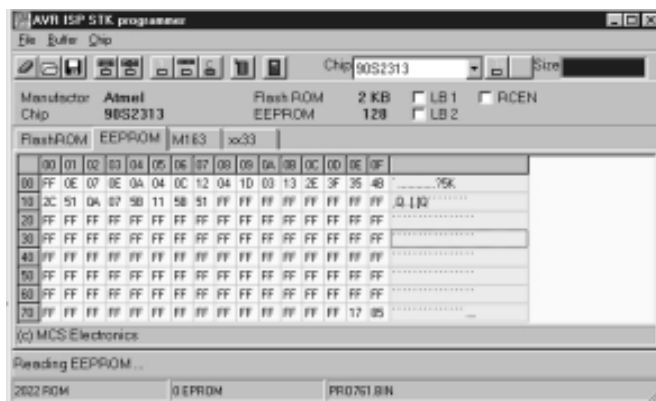
Do wprowadzania danych zastosowano obrotowy impulsator, który podczas obracania jego osią generuje impulsy na dwóch wyjściach. Kolejność występowania tych impulsów jest taka, że dołączony do wyjść impulsatora procesor może z łatwością nie tylko je zliczyć, ale także określić kierunek obrotu osi impulsatora.

Wewnętrzny oscylator procesora stabilizowany jest rezonatorem kwarcowym o częstotliwości podstawowej 8MHz. Jako element wykonawczy, dołączony do złącza CON2, zastosowano przełącznik RL1, którego cewka zasilana jest przez tranzystor T3,ysterowywany z wyjścia PD6

procesora. Złącze CON3 służy do dołączenia do układu serwomechanizmu modelarskiego.

Układ zasilany jest napięciem stałym o wartości około 12VDC, doprowadzonym do złącza CON1. Napięcie to jest następnie obniżane i stabilizowane na poziomie 5VDC za pomocą scalonego stabilizatora napięcia IC1. Układ IC1 zapewnia pewny start procesora po włączeniu zasilania i zabezpiecza przed niekontrolowanym działaniem procesora w przypadku obniżenia napięcia zasilania poniżej dopuszczalnej wartości.

Analizę działania układu rozpocniemy od momentu pierwszego włączenia napięcia zasilającego. Pierwszą czynnością, jaką wy-



Rys. 2. Zawartość pamięci danych EEPROM podczas działania podprogramu rejestracji kodu.

```

List. 1.
Sub Numbers_registration      'podprogram rejestracji kodu dostępu
  Eeprom_address = 1         'wstępne ustawienie adresu pamięci danych EEPROM na 1
  For Temp1 = 1 To 10
    Flash_flag = Not Flash_flag 'zmień wartość zmiennej pomocniczej FLASH_FLAG na
                                'przeciwną. Efekt działania tego fragmentu programu możemy
                                'prześledzić analizując podprogram sterowania multipleksowanym
                                'wyświetlaniem danych

    Waitms 100
    Next Temp1

  Do
    Switch_scan               'wezwiń podprogram badania stanu styków impulsatora. Procedura ta
                                'została omówiona w EP5/2001 (regulator obrotów silników DC)
    If Button_flag = 1 Then 'jeżeli został naciśnięty przycisk impulsatora, to:
      Writeeeprom Disp, Eeprom_address 'zapisz w pamięci danych EEPROM aktualnie
                                        'prezentowaną na wyświetlaczach liczbę
      Flash_flag = 0                 'włącz wyświetlanie kropek dziesiętnych na wyświetlaczach LED
      Waitms 100
      Flash_flag = 1                 'wyłącz wyświetlanie kropek dziesiętnych na wyświetlaczach LED
      Wait 1
      Set Portd.3                    'spróbuj ustawić stan wysoki na przycisku impulsatora
      If Pind.3 = 0 Then             'jeżeli próba nieudana, co oznacza że przycisk jest naciśnięty
                                      'dłużej niż 1 sekundę, to:
        Writeeeprom Eeprom_address, 126 'zapisz w pamięci danych EEPROM informacje
                                          'o ilości wprowadzonych liczb
        Temp1 = 133                  'zmienna pomocnicza TEMP przyjmuje wartość 133
                                      '(patrz opis procedury badania stanu pamięci EEPROM)
        Writeeeprom Temp1, 127        'zapisz tą wartość pod adresem 127
        Waiting                       'wezwiń podprogram oczekiwania na podanie kodu dostępu
      End If
      Incr Eeprom_address 'zwiększ wartość adresu pamięci EEPROM o 1
    End If
  Loop
End Sub

```

konuje procesor jest sprawdzenie, czy w pamięci danych EEPROM znajdują się liczby, wchodzące w skład klucza:

```

Readeeprom Temp, 127
If Temp <> 133 Then
  Numbers_registration
EndIf

```

Całkowicie umowna liczba „133” jest zapisywana w pamięci EEPROM pod adresem 127 po zarejestrowaniu kodu klucza. Stwierdzenie innej wartości pod tym adresem powoduje automatyczne przejście do podprogramu rejestracji kodu klucza otwierającego zamek (list. 1).

Efekt działania podprogramu rejestracji klucza można zobaczyć na rys. 2, gdzie pokazana została zawartość pamięci danych EEPROM. Zostały tam zapisane daty bitów pod Grunwaldem i Austerlitz, data urodzin autora tego projektu i kilka innych liczb. Szczerze polecam taką metodę „komponowania” kodu klucza, ułatwiającą jego zapamiętanie.

W układzie zamka zastosowano multipleksowane wyświetlanie danych, realizowane z częstotliwością ok. 488Hz. Multipleksowanie taktowane jest za pomocą *timera0*, którego konfigurację sprzętową i programową określają polecenia:

```

Config Timer0=Timer, Prescale=64
On Timer0 Multiplexing

```

Wyświetlanie danych realizowane jest za pomocą podprogra-

mu MULTIPLEXING, wywołwanego przez przerwanie pochodzące od timera0 (list. 2).

Po zarejestrowaniu kodu klucza dostępu, podobnie jak po otwarciu i ponownym zamknięciu zamka program przechodzi do podprogramu oczekiwania WAITING. Nie dzieje się tam nic szczególnego, ponieważ jedyną funkcją wykonywaną przez program jest sprawdzanie, czy przycisk impulsatora nie został przypadkiem naciśnięty oraz zerowanie sprzętowego watchdoga.

```

Sub Waiting
'podprogram oczekiwania na
'wybranie kodu dostępu
Disable Timer0
'zawieś obsługę przerwania
'timera0
Set Portd.0: Set Portd.1
'wyłącz wyświetlacze
Wait 1 'zaczekaj sekundę
Do
Set Portd.3

```

```

List. 2.
Multiplexing:
Set Portd.0: Set Portd.1 'wstępnie wyłącz obydwa wyświetlacze
Portb = 255              'wyłącz wszystkie segmenty wyświetlaczy
Digit_number = Not Digit_number 'zmień numer aktywnego wyświetlacza
If Digit_number = 0 Then 'jeżeli aktywny będzie wyświetlacz 0, to:
  Temp = Disp / 10       'oblicz wartość dziesiątek wyświetlanej liczby
  Portb = Lookup(temp, 7segment) 'przekodowanie otrzymanej wartości na kod wyświetlacza
                              'siedmiosegmentowego
  If Flash_flag = 1 Then Set Portb.7 'jeżeli punkty dziesiętne wyświetlaczy nie mają
                                      'migotać, to ustaw stan wysoki na pinie 7 portu B
  Reset Portd.1           'włącz pierwszy wyświetlacz
End If

If Digit_number = 1 Then 'jeżeli aktywny będzie drugi wyświetlacz, to:
  Temp = Disp / 10       'oblicz wartość jednostek wyświetlanej liczby
  Temp = Temp * 10
  Temp = Disp - Temp
  Portb = Lookup(temp, 7segment) 'przekodowanie otrzymanej wartości na kod wyświetlacza
                              'siedmiosegmentowego
  If Flash_flag = 1 Then Set Portb.7 'jeżeli punkty dziesiętne wyświetlaczy nie mają
                                      'migotać, to ustaw stan wysoki na pinie 7 portu B
  Reset Portd.0         'włącz drugi wyświetlacz
End If

Return
.....
7segment:
Data 64, 121, 36, 48, 25, 18, 2, 120, 0, 16

```

```

'sprawdź stan przycisku
'impulsatora
If Pind.3 = 0 Then
'jeżeli przycisk został
'naciśnięty, to:
  Verification
EndIf

Reset Watchdog
Loop
End Sub

```

Naciśnięcie przycisku impulsatora powoduje natychmiastowe przejście do podprogramu identyfikacji wprowadzanego kodu klucza. Włączone zostają wyświetlacze i trzykrotnym błyskiem kropek dziesiętnych jest sygnalizowana gotowość układu do przyjmowania wprowadzanych liczb. Dla wygody układ startuje z początkową wartością równą 50 (list. 3).

Montaż i uruchomienie

Na rys. 3 pokazano rozmieszczenie elementów na płytce drukowanej, wykonanej na laminacie dwustronnym z metalizacją otworów. Montaż tak prostego układu nie wymaga chyba szczegółowych komentarzy, z wyjątkiem jednego: wyświetlacze siedmiosegmentowe LED oraz impulsator obrotowy muszą zostać zamontowane od umownej strony ścieżek, czyli od strony lutowania. Pozwoli to na łatwiejsze umieszczenie układu w obudowie.

Zmontowany ze sprawdzonych elementów układ zamka nie wymaga regulacji i po zaprogramowaniu kodu klucza nadaje się do eksploatacji. Po pierwszym włączeniu zasilania układ automatycznie przechodzi do trybu rejestracji kodu klucza dostępu, o czym jes-

List. 3.

```

Sub Verification
Disp = 50          'ustaw wstępną wartość wyświetlanej liczby na 50
Enable Timer0     'włącz wyświetlacz
For Temp1 = 1 To 3
  Flash_flag = Not Flash_flag 'zmień wartość zmiennej pomocniczej FLASH_FLAG na
                              'przeciwną. Efekt działania tego fragmentu programu możemy prześledzić
                              'analizując podprogram sterowania multipleksowanym wyświetlaniem danych
  Waitms 200
Next Temp1

Time_counter = 0  'wyzeruj licznik upływu czasu
Eeprom_address = 1 'ustaw wstępnie adres pamięci danych EEPROM na 1
Code_flag = 1     'zakładamy, że wprowadzany kod jest poprawny
Readeeprom Numbers, 126 'odczytaj z pamięci EEPROM ilość liczb w kodzie

Do
  Switch_scan 'wezij podprogram badania stanu styków impulsatora. Procedura ta
              'została omówiona w EP5/2001 (regulator obrotów silników DC)
  If Button_flag = 1 Then 'jeżeli został naciśnięty przycisk impulsatora, to:
    Time_counter = 0      'wyzeruj licznik upływu czasu
    Readeeprom Temp1, Eeprom_address 'odczytaj z pamięci EEPROM wartość kolejnej liczby kodu

    If Temp1 <> Disp Then Code_flag = 0
    'jeżeli odczytana wartość nie jest równa wartości wprowadzonej, to wyzeruj zmienną
    'poprawności kodu. Od tego momentu wprowadzanie dalszych liczb nie odnosi już żadnego
    'skutku

    If Code_flag = 1 And Eeprom_address = Numbers Then Opening
    'jeżeli wszystkie wprowadzone liczby są poprawne, zostały podane w prawidłowej
    'kolejności i żadna liczba nie została do kodu dodana, to wezwij podprogram otwierania
    'zamka

    Incr Eeprom_address 'zwiększ wartość licznika adresu pamięci EEPROM o 1
    Flash_flag = 0      'wygeneruj krótki błysk punktów dziesiętnych wyświetlaczy
    Waitms 255
    Flash_flag = 1
  End If

  If Time_counter > 12000 Then Exit Do
  'jeżeli licznik czasu osiągnął wartość 12000, to powróć do stanu oczekiwania
Loop
End Sub

```

teśmy zawiadomiani dziesięcioma błyskami kropek dziesiętnych na wyświetlaczach. Kolejność postępowania podczas programowania kodu klucza jest następująca:

1. Za pomocą pokrętki impulsatora ustawiamy na wyświetlaczach pierwszą liczbę wchodzącą w skład kodu.

2. Krótkie naciśnięcie przycisku impulsatora powoduje wprowadzenie ustawionej liczby do pamięci i automatyczne przejście do programowania kolejnej liczby. Potwierdzone jest to krótkim błyskiem kropek dziesiętnych na wyświetlaczach.

3. Czynności opisane w punktach 1 i 2 powtarzamy tyle razy, ile liczb mamy zamiar zaprogramować. Może ich być maksymalnie 125.

4. Po ustawieniu na wyświetlaczach ostatniej liczby kodu klucza przytrzymujemy przycisk dłużej niż uprzednio, przez okres około 1 sekundy. Spowoduje to zapisanie do pamięci ostatniej liczby, wyłączenie wyświetlaczy i przejście układu w stan czuwania, o czym zostajemy zawiadomieni za pomocą pięciu błysków punktów dziesiętnych na wyświetlaczach.

Otwieranie zamka przebiega w sposób podobny do jego programowania. Pierwsze naciśnięcie przycisku impulsatora powoduje włączenie wyświetlaczy i przygotowanie układu do wprowadzenia kodu klucza. Za pomocą pokrętki impulsatora ustawiamy na wyświetlaczach kolejne liczby kodu i potwierdzamy to krótkim naciśnięciem przycisku. Następnie, w identyczny sposób wprowadzamy kolejne liczby. Jeżeli wszystkie liczby wchodzące w skład kodu zostaną wprowadzone we właściwej kolejności, to zamek otworzy się natychmiast po podaniu ostatniej liczby. Zamek pozostanie otwarty aż do ponownego naciśnięcia przycisku impulsatora.

Wprowadzenie błędnej liczby lub liczby wchodzącej w skład

WYKAZ ELEMENTÓW

Rezystory

R1, R2: 3,3kΩ

R3, R4, R5: 1kΩ

Kondensatory

C1: 470μF/16V

C2, C3: 100nF

C4: 100μF/10V

C5, C6: 27pF

Półprzewodniki

D1: 1N4148

IC1: 7805

IC2: AT90S2313 (zaprogramowany)

IC3: DS1813

T1, T2: BC557

T3: BS107

Różne

CON1: ARK2 (3,5mm)

CON2: ARK3

CON3: 3 x goldpin

DP1, DP2: wyświetlacz siedmiosegmentowy LED, wsp. anoda

Q1: rezonator kwarcowy 8MHz

RL1: przekaźnik RM96

S1: impulsator obrotowy

kodu klucza, ale podanej w niewłaściwej kolejności powoduje unieważnienie całej operacji wprowadzania kodu, a otwarcie zamka staje się niemożliwe. Należy wtedy odczekać ok. 20 sekund, po czym wyświetlacze zostaną wyłączone, a my możemy rozpocząć kolejną próbę wprowadzenia poprawnego klucza.

Podczas eksploatacji zamka może zajść konieczność zmiany ustawionego uprzednio kodu klucza. Ponowne przejście w tryb rejestracji kodu wymuszamy następująco:

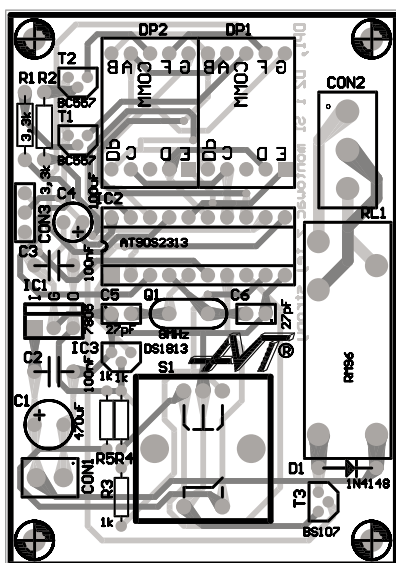
1. Wyłączamy zasilanie zamka.
2. Naciskamy przycisk impulsatora

3. Trzymając cały czas przycisk impulsatora naciśnięty ponownie włączamy zasilanie.

Po wykonaniu tych czynności układ automatycznie przejdzie w tryb rejestracji nowego kodu dostępu.

Zbigniew Raabe
zbigniew.raabe@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/styczen02.htm> oraz na płycie CD-EP01/2002B w katalogu PCB.



Rys. 3. Rozmieszczenie elementów na płytce drukowanej.