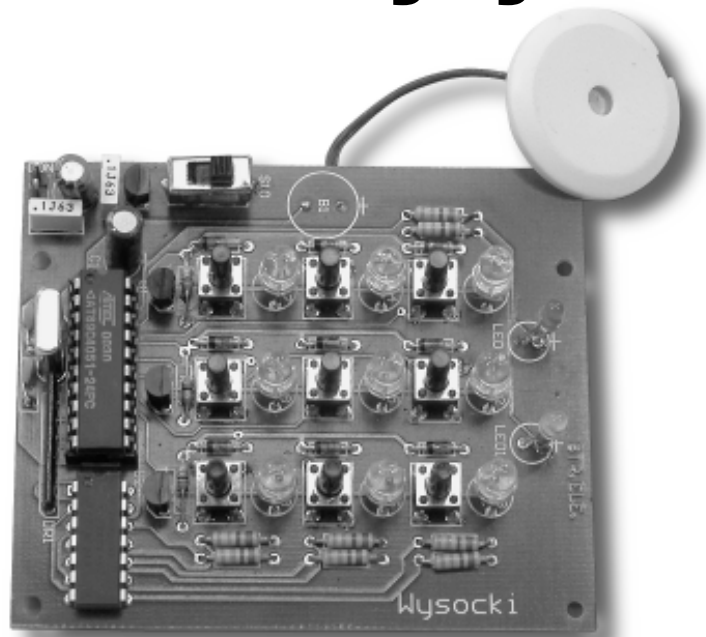


Gra w kółko i krzyżyk

AVT-5101

Przedstawiamy łatwy w wykonaniu układ elektronicznej wersji jednej z najpopularniejszych gier towarzyskich: kółko i krzyżyk.

Rekomendacje: grę polecamy szczególnie uczniom i studentom, którym pomoże ona zabić lekcyjną (wykładową) nudę, a także wszystkim fanom dobrej rozrywki.



Początkowo planowałem wykonanie kolejnego wcielenia gry w kości, ale po przemyśleniu uznałem, że temat jest tak wyeksploatowany, że zapewne niewielu Czytelników poświęciłoby uwagę takiemu opracowaniu. O dziwo nigdy nie spotkałem się z elektroniczną wersją popularnej towarzyskiej gry w kółko i krzyżyk. Pierwszym odruchem była chęć zastosowania wyświetlacza graficznego, lecz koszt wykonania takiej zabawki byłby nieporównywalnie duży w stosunku do przydatności. Ponieważ w grze mamy do postawienia tylko dwa możliwe symbole, najprostszym rozwiązaniem okazało się użycie matrycy 3x3 diod dwukolorowych wraz z zestawem przycisków. Program umożliwia grę w dwójkę oraz z urządzeniem. Mamy do wyboru 3 poziomy trudności. Gramy do 3 zwycięstw, remisy nie dają punktów nikomu.

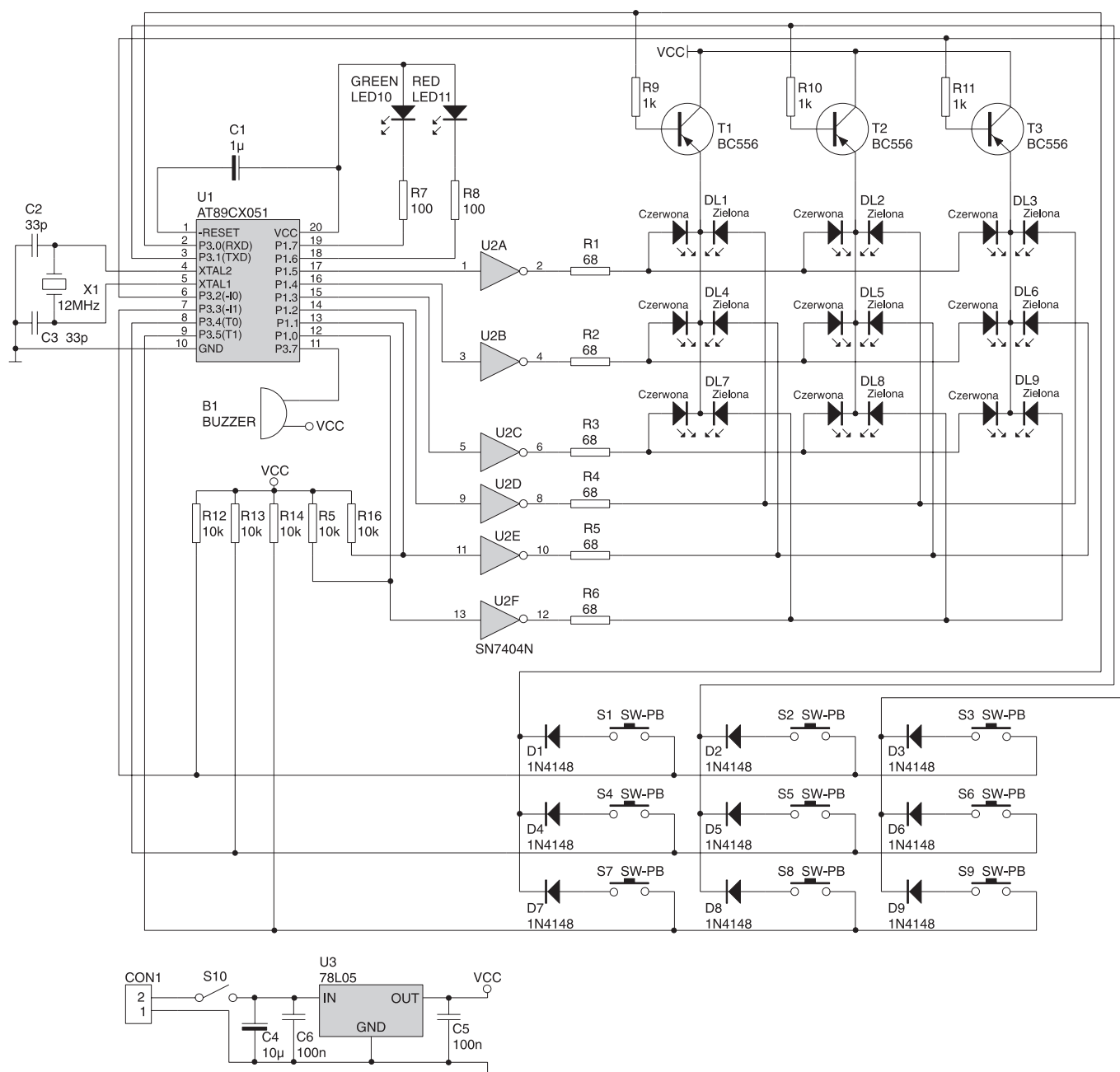
Zachęcam wszystkich do wykonania tej gry - na pewno uchroni nas przed stosem pokreślonych kartek i zapewni rozrywkę np. na nudnych lekcjach czy wykładach.

Opis działania

Układ sterowania sprowadza się do nieśmiertelnego procesora 89C4051, sześciu inwerterów służących za bufory wyjściowe oraz

3 tranzystorów. Na **rys. 1** pokazano schemat elektryczny urządzenia. Mikrokontroler pracuje w typowej dla siebie aplikacji. Kondensator C1 ładuje się po włączeniu zasilania, zapewniając zerowanie mikrokontrolera. Do zasilania całości służy miniaturowy stabilizator napięcia 78L05. Układ został przewidziany do zasilania z baterii 9 V.

Bliższego omówienia wymaga część sterująca diodami LED i przyciskami. Matryca jest sterowana multipleksowo co oznacza, że tylko jedna kolumna LED-ów jest zapalona w danym momencie. Jednocześnie podczas „przemiatania“ matrycy diod, skanowana jest klawiatura. Procesor podaje stany 0 na bazy kolejnych tranzystorów włączając kolumnę trzech diod. Następnie z drugiej strony poprzez inwertery zapewniające odpowiednio dużą wydajność prądową (porty procesora w stanie 1 mają ją bardzo małą) zapalane są wybrane diody. Cały cykl powtarza się na tyle często, że mamy wrażenie ciągłego świecenia wszystkich LED-ów. Gdy na bazie danego tranzystora występuje 0 i w danej kolumnie naciśnięty jest przycisk zaczyna przewodzić dioda i port mikrokontrolera jest zwierany do masy. Diody są niezbędne, aby naciśnięcie przycisku nie zwierano linii na stałe



Rys. 1. Schemat elektryczny gry

do „+“ zasilania, gdy nie jest ona wybrana. Oporniki podciągające R12...R14 dodano profilaktycznie, aby uniknąć zakłóceń. Z kolei R15...R16 są wymagane do poprawnej pracy układu jeśli zastosujemy inwertery w wersji CMOS (porty P1.0 i P1.1 są typu otwarty kolektor).

Na płytce drukowanej zamiast 5 rezystorów zastosowano R-Pack. Rezystory R1...R6 ograniczają prąd diod do ok. 10 mA, co zapewnia wystarczającą jasność i maksymalny pobór prądu z baterii ok. 30 mA. Dodatkowym osprzętem jest przetwornik piezoceramiczny z wbudowanym generatorem oraz

dwie diody świecące - czerwona i zielona, które wskazują gracza. Zamiast buzzera można zastosować membranę piezo, lecz generowany dźwięk jest niskiej jakości z powodu cyklicznie wywoływanej procedury odświeżania matrycy. Odpowiednio zmodyfikowane wersje programów można znaleźć na płycie CD-EP2/2003B.

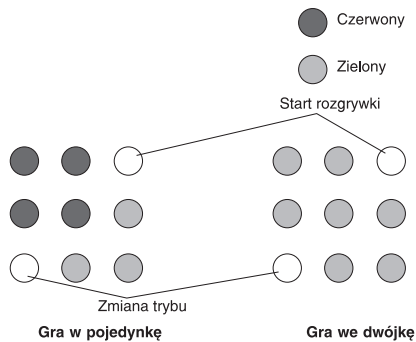
Działanie programu, obsługa gry

Opogramowanie powstało w Bascom ie 8051. Całkowita wielkość kodu to ok. 3500 bajtów. Jest to dość dużo, ale wynika to po części ze słabych optymalizacji

niektórych funkcji przez kompilator i braku obsługi zmiennych tablicowych dwuwymiarowych, które znacznie ułatwiły by np. sprawdzanie czy któryś z graczy stworzył linię.

Zaraz po uruchomieniu jest sprawdzane, czy któryś z klawiszy nie jest wciśnięty. Jeśli tak, to zostaje całkowicie wyłączony buzzer, co umożliwia nam granie nie zwracając na siebie uwagi ciągłym pikaniem.

Do wyboru grę w jednym z dwóch trybów: *single player* oraz *dual player*. Aby zmienić bieżący tryb pracy należy nacisnąć przycisk pod diodą w dolnym



Rys. 2. Konfiguracja gry

lewym rogu, a następnie przycisk przeciwny w celu rozpoczęcia rozgrywki (rys. 2).

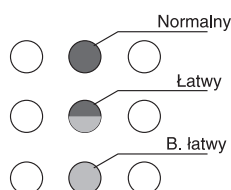
Gdy gramy z urządzeniem musimy wybrać stopień trudności gry. Żeby to zrobić naciskamy jeden z przycisków w środkowej kolumnie (rys. 3).

Sztuczna „inteligencja” naszej zabawki jest stosunkowo ograniczona:

- tryb bardzo łatwy: program sprawdza tylko czy może zrobić ruch tworzący linię, jeśli nie - losuje,
- tryb łatwy: jw., sprawdza czy może zablokować zwycięstwo przeciwnika, jeśli tak to losuje czy ma to zrobić,
- tryb normalny: jw., zawsze blokuje kombinacje przeciwnika umożliwiające wygraną.

Po wybraniu ustawień zaczyna się rozgrywka. Zawsze pierwszy jest gracz czerwony. Migająca dioda ponad polem gry informuje który gracz musi teraz wykonać ruch. Przykładowa pole gry pokazano na rys. 4. W takim wypadku, po wybraniu przez czerwonego gracza pola w górnym prawym rogu, zwycięska kombinacja mignie 3 razy i program przedstawi nam punktację, jak przykładowo pokazano to na rys. 5.

Remis jest sygnalizowany 3 piknięciami i nikomu nie jest przyznawany punkt. W lewej kolumnie znajduje się punktacja gracza zielonego, a w prawej czerwonego. Teraz naciśnięcie dowolnego przycisku spowoduje przejście do następ-

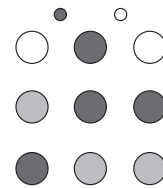


Rys. 3. Wybór stopnia trudności gry

nej partii. Zawsze partię rozpoczyna zwycięzca z poprzedniej. Gdy wystąpił remis gracz inicjujący jest losowany. Jeśli któryś gracz wygrał po raz 3, to po wyświetleniu wyniku całe pole zapali się na kolor gracza i mignie 3 razy. Naciśnięcie dowolnego klawisza spowoduje powrót do menu wyboru trybu gry.

Oprogramowanie wykorzystuje do odświeżania wyświetlacza wewnętrzny timer procesora 8051. Nie będę przytaczał całego programu, ale procedura wyświetlająca (list. 1) jest o tyle godna uwagi, że uzupełni wiedzę o działaniu sprzętu.

W programie najważniejszą rolę pełnią dwie zmienne tablicowe L(x) oraz S(x). Zawierają one informacje o tym, które diody mają być zapalone (L) oraz które przyciski są naciśnięte. Cała procedura jest ujęta w warunku IF sprawdzającym ile razy zostało wywołane przerwanie. Służy to zmniejszeniu częstotliwości skanowania. Następnie, w zależności od wartości zmiennej refresh, program przepisuje zawartość tablicy L(x) do tablicy TL(x). Jest to przydatne gdy nasza sztuczna „inteligencja” sprawdza różne możliwe kombinacje - poprzez wyłączenie przepisywania na ekranie nie będzie widać „śmieci”. Wszystkie wyjścia przechodzą w stan nieaktywny. Następnie program w zależności od zawartości zmiennych P11 i P12 zapala diody umieszczone ponad polem gry. Wewnątrz warunku jest umieszczona swego rodzaju pętla, dzięki której stan diody jest negowany po ok. 200 odświeżeniach - zapewnią to migotanie z częstotliwością ok. 1 Hz. Kolejny warunek sprawdza, czy nie przekroczyliśmy dopuszczalnej wartości licznika kolumn. W zależności od kolumny zmiennym v1, v2, v3 oraz vcol przypisywane są odpowiednie wartości. Zmienne vx zawierają numery pozycji z tablicy do wyświetlenia, a vcol wartość sterującą tranzystorami kolumn. W wartościach tablicy L(x) bit 0 steruje diodą zieloną, a bit 1 czerwoną. Wartości są przepisywane na odpowiednie wyprowadzenia portu P1. Następnie włączany jest tranzystor kolumny (odpowiednia wartość w vcol).



Rys. 4. Wygląd przykładowego pola gry

Dalej program sprawdza, czy któryś z przycisków nie jest wciśnięty. Jeśli tak - ustawia odpowiedni element tablicy S(x) na 1.

Pewnym problemem było wykrycie, czy gracz ułożył zwycięską kombinację, ponieważ Bascom nie

List. 1. Procedura obsługująca wyświetlanie matrycowe

```

Scan:
Incr Tim

If Tim = 3 Then
  Tim = 0

  If Refresh = 1 Then
    For Rtemp = 1 To 9
      Tl(rtemp) = L(rtemp)
    Next Rtemp
  End If

  P1 = P1 Or &B00111111
  P3 = P3 Or &B01111111
  If P11 = 1 Then
    Incr Tim1
    If Tim1 = 250 Then
      P1.7 = Not P1.7
      Tim1 = 0
    End If
  Else
    P1.7 = 1
  End If
  If P12 = 1 Then
    Incr Tim2
    If Tim2 = 250 Then
      P1.6 = Not P1.6
      Tim2 = 0
    End If
  Else
    P1.6 = 1
  End If
  If Col = 4 Then Col = 1
  Select Case Col

  Case 1:
    V1 = 1
    V2 = 2
    V3 = 3
    Vcol = &B01111110

  Case 2:
    V1 = 4
    V2 = 5
    V3 = 6
    Vcol = &B01111101

  Case 3:
    V1 = 7
    V2 = 8
    V3 = 9
    Vcol = &B01111011
  End Select

  P1.0 = Not Tl(v1).0
  P1.1 = Not Tl(v2).0
  P1.2 = Not Tl(v3).0
  P1.3 = Not Tl(v1).1
  P1.4 = Not Tl(v2).1
  P1.5 = Not Tl(v3).1

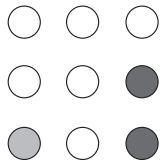
  P3.0 = 0
  P3.1 = 0
  P3.2 = 0

  P3 = P3 Or Vcol

  Temp = P3 And &B00111000
  Select Case Temp
  Case &B00011000: S(v1) = 1
  Case &B00101000: S(v2) = 1
  Case &B00110000: S(v3) = 1
  Case Else:
    S(v1) = 0
    S(v2) = 0
    S(v3) = 0
  End Select

  Incr Col
End If
Return

```



Rys. 5. W ten sposób jest sygnalizowana punktacja gry

pozwala na tworzenie tablic dwuwymiarowych. Postanowiłem więc zastosować sprawdzanie wszystkich możliwych kombinacji. Zachęcam zainteresowanych do przejrzenia całego programu - do ściągnięcia z Internetu, dostępny jest także na płycie CD-EP2/2003B.

Małego wyjaśnienia wymaga jeszcze procedura obsługi buzzera, którą przedstawiono na **list. 2**.

Zmienna *beepoff* zostaje ustawiona, gdy przytrzymamy przy starcie jakiś klawisz (patrz wy-

List. 2. Procedura obsługi sygnalizatora akustycznego

```
Sub Beep()
  If Beepoff = 0 Then
    For Beepx = 1 To 20
      P3.7 = 0
      Delay
      P3.7 = 1
      Delay
    Next Beepx
  End If
End Sub
```

żej). Pętla *for* tworzy prosty generator o częstotliwości teoretycznie 5 kHz (opóźnienie *delay* - ok. 100 μ s), lecz jest to w praktyce znacznie mniej, ponieważ praca jest przerywana przez procedurę odświeżania wyświetlacza i dochodzą opóźnienia pętli. W przypadku zastosowania gotowego generatora z przetwornikiem piezoceramicznym procedura upraszcza się do postaci pokazanej na **list. 3**.

Montaż i uruchomienie

Urządzenie zmontowano na płytce drukowanej o wymiarach ok. 70 x 80 mm, której schemat montażowy pokazano na **rys. 6**. Jest ona przystosowana do obudowy KM33B (ważne - wersja bez litery B w oznaczeniu ma maskownicę do głośnika i nie nadaje się do naszego układu). W obudowie przewidziano miejsce na baterię 9V.

Pewien problem stanowi to, że standardowo płytka jest znacznie odsunięta od przedniej ścianki obudowy - należy zastosować mikro-

WYKAZ ELEMENTÓW

Rezystory

R1...R6: 68 Ω
R7, R8: 100 Ω
R9, R10, R11: 1k Ω
R12...R16: 10k Ω

Kondensatory

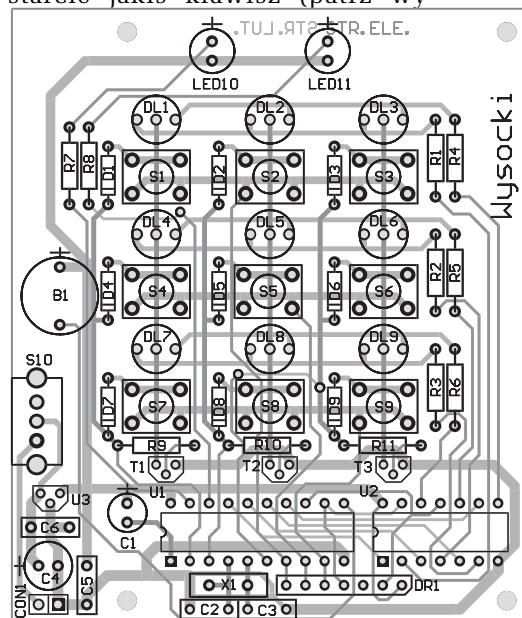
C1: 1 μ F/16V
C2, C3: 33pF
C4: 10 μ F/16V
C5, C6: 100nF

Półprzewodniki

D1...D9: 1N4148
DL1...DL9, LED11: LED czerwona
LED10: LED zielona
T1...T3: BC556
U1: AT89CX051 (zaprogramowany)
U2: SN7404N
U3: 78L05

Różne

B1: BUZZER
X1: 12MHz
S1...S9: SW-PB



Rys. 6. Rozmieszczenie elementów na płytce drukowanej

List. 3. Procedura obsługi generatora akustycznego

```
Sub Beep()
  If Beepoff = 0 Then
    P3.7 = 0
    Waitms 10
    P3.7 = 1
  End If
End Sub
```

przełączniki z przyciskami o długości 9 mm, diody wlotować tak aby były na równi z „łepkami“ przycisków.

Po dopasowaniu otworów w przedniej ściance, musimy przyciąć kołki wewnątrz obudowy i dobrać odpowiednio długie wkręty do skręcenia całości. Żeby płytka nie spadała do wnętrza obudowy można zastosować odpowiedniej długości tulejki dystansowe lub „na drodze“ wkręta wstawić ucięte fragmenty kołków.

Układ po złożeniu i zaprogramowaniu procesora działa od razu (w zależności od zastosowanego kondensatora C1 może startować dopiero po np. sekundzie).

Michał Wysocki
msoft@satkabel.com.pl
www.msoft.prv.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/luty03.htm> oraz na płycie CD-EP2/2003B w katalogu PCB.