

Pilot komputerowy

μP-3007



Czy ktokolwiek z Was, użytkowników komputera, nie odczuwał nigdy potrzeby sterowania swoim PC-tem z pewnej odległości, np. z łóżka czy biurka? Czy nigdy nie musieliście podchodzić do komputera tylko po to, żeby przełączyć słuchaną "empetrójkę" lub zmienić kanał TV? Przykładów zastosowania pilota komputerowego można wymieniać wiele. Wykorzystanie go do różnych celów, aby sterować komputerem za pomocą dowolnego pilota RC-5. Co ważne, jest to przystawka, która nie zakłóca normalnej pracy klawiatury.

Opis układu

Omawiane urządzenie jest dość nietypowe, mianowicie udaje klawiaturę po otrzymaniu odpowiednich instrukcji od pilota. Jak zwykle w przypadku stosowania techniki mikroprocesorowej, układ pod względem budowy jest bardzo prosty. Jego schemat ideowy przedstawiony został na **rysunku 1**. Cała inteligencja mieści się w głównym układzie

scalonym, którym jest mikrokontroler AT89C2051. Układ zasilany jest z linii klawiatury, dzięki czemu nie potrzeba zewnętrznego zasilania. Serce układu (U1) komunikuje się z pamięcią szeregową (U2) po szynie I²C, odbiera sygnały od scalonego odbiornika podczerwieni (U3) oraz steruje czterema tranzystorami (T1-T4) potrzebnymi przy transmisji i dwoma diodami (D1, D2) sygnalizującymi, czy sygnał z pilota został zinterpretowany, czy odrzucony.

Nasz układ to jednak nie wszystko, czego potrzebować będziemy do sterowania komputerem. Niezbędny będzie jeszcze pilot. No właśnie, pilot, ale jaki? Dowolny pilot RC-5 lub "Minipilot RC5", kit AVT-2427 opisany w EdW 9/2000 na stronie 77. Ma on wystarczającą liczbę przycisków (16) i jest stosunkowo tani. Poza tym dokładnie wiemy, jaki kod odpowiada danemu przyciskowi (komendy podane zostały na **rysunku 2**, adres dla wszystkich przycisków równy jest 0), w przeciwieństwie do fabrycznych pilotów uniwersalnych, które potrafią zachowywać się różnie.

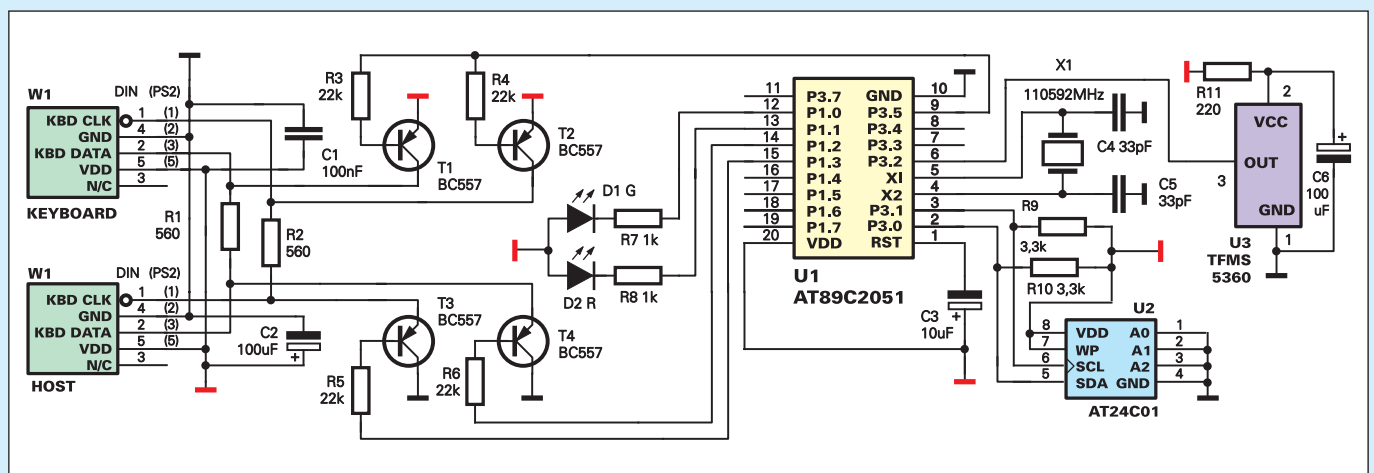
Po zaznajomieniu się ze schematem, zajmijmy się pracą układu "od kuchni".

Komenda odebrana od pilota za pośrednictwem układu U3 wpisywana jest do pamięci szeregowej jako adres, z którego odczytywany jest bajt. Bajt ten nie jest niczym innym jak kodem odpowiadającym danemu przyciskowi klawiatury, którego wciśnięcie

7	8	9	F
⓪7	⓪8	⓪9	⓪D
4	5	6	E
⓪4	⓪5	⓪6	⓪C
1	2	3	D
⓪1	⓪2	⓪3	⓪B
0	A	B	C
⓪0	⓪28	⓪29	⓪A

Rys. 2

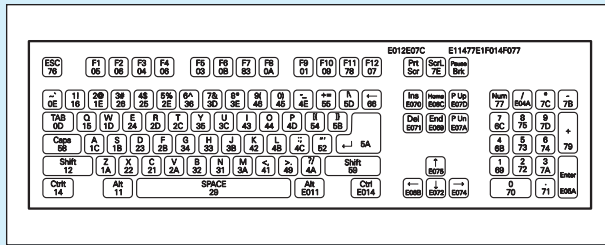
Rys. 1



symulować miała obsługa pilota. Właściwie wiemy już skąd układ zna kod klawisza, ale przecież niektóre przyciski (np. "strzałki") są klawiszami specjalnymi i aby je wywołać, trzeba przed ich kodem wysłać komputerowi poprzednik klawisza specjalnym (szesnastkowo E0). Problem ten można rozwiązać całkiem prosto. Ponieważ pilot może przesłać komendy od 0-63, połowa pamięci U2 leży odłogiem. Nic nie stoi na przeszkodzie, aby wykorzystać tę część pamięci na rzecz klawiszy specjalnych. W tym celu,

w nieużywanej dotąd przestrzeni adresowej od 63 do 127, zapisujemy kody odpowiadające klawiszom specjalnym pod adresami będącymi komendą powiększoną o 64. Rozpatrzmy teraz zachowanie układu, gdy otrzyma on komendę dotyczącą klawisza specjalnego. Najpierw mikrokontroler sprawdza, czy pod adresem odpowiadającym komendzie znajduje się jakiś kod. Kiedy upewni się, że kodu nie ma (odczytał znak pusty - szesnastkowo FF), to sprawdza, czy pod adresem o 64 wyższym coś jest. Jeśli jest, to już wie, że ma to być klawisz specjalny i najpierw przesyła komputerowi poprzednik klawisza specjalnego, a dopiero potem kod klawisza odczytany z danego adresu. **Rysunek 3** przedstawia kody wszystkich klawiszy klawiatury, z których możemy wybrać potrzebne i wpisać do pamięci U2 według szablonu z **rysunku 4** pod adresy odpowiadające wybranym przyciskom pilota. Jeśli jakiś przycisk ma być nieużywany, to w miejsce kodu klawisza musimy wpisać szesnastkowo FF.

Mając pewne pojęcie na temat identyfikacji klawisza, możemy przyrzeć się bliżej transmisji. Jak już wcześniej wspomniałem, mikrokontroler steruje kilkoma tranzystorami. To one właśnie zarządzają szyną danych (DTA) i zegara (CLK) komputera, którymi zwykle "zajmuje się" klawiatura. Można powiedzieć, że nasze urządzenie wtrąca się pomiędzy klawiaturę i komputer. Takie brutalne działanie jest nieuniknione, ponieważ linia klawiatury przystosowana jest tylko do jednego odbiornika i jednego nadajnika. Jak w takim razie przystosować transmisję do naszych potrzeb? Całkiem prosto! Szeregowo w linie musimy włączyć rezystory o niedużej oporności (R1 i R2). Teraz wystarczy już tylko po stronie klawiatury "podciągnąć" linię do napięcia zasilania (T1, T2), a po stronie komputera, w takt transmisji, zwierać końcówkę rezystora do masy (T3,



Rys. 3

000X)	0	1	2	3	4	5	6	7	-	8	9	A	B	C	D	E	F
0000)	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	-	(8)	(9)	(C)	(D)	(E)	(F)	FF	FF
0010)	FF	FF	FF	FF	FF	FF	FF	FF	-	FF	FF	FF	FF	FF	FF	FF	FF
0020)	FF	FF	FF	FF	FF	FF	FF	FF	-	(A)	(B)	FF	FF	FF	FF	FF	FF
0030)	FF	FF	FF	FF	FF	FF	FF	FF	-	FF	FF	FF	FF	FF	FF	FF	FF
0040)	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	-	(8)	(9)	(C)	(D)	(E)	(F)	FF	FF
0050)	FF	FF	FF	FF	FF	FF	FF	FF	-	FF	FF	FF	FF	FF	FF	FF	FF
0060)	FF	FF	FF	FF	FF	FF	FF	FF	-	(A)	(B)	FF	FF	FF	FF	FF	FF
0070)	FF	FF	FF	FF	FF	FF	FF	FF	-	FF	FF	FF	FF	FF	FF	FF	FF

Rys. 4

T4). Ale czy aby na pewno taki zabieg jest konieczny? Przecież te linie sterowane są za pomocą bramek typu otwarty kolektor, więc czy nie można po prostu zwierać danej linii do masy? Odpowiedz na to pytanie i trochę więcej szczegółów dotyczących transmisji (potrzebnych do napisania programu) znaleźć można w EdW 1/2001 na stronie 25. Żeby się zbytnio nie powtarzać, przypomnę jedynie, że linia klawiatury jest linią dwukierunkową. Wynika z tego, że również komputer może przysłać rozkazy klawiaturze (np. załącz/ wyłącz kontrolkę Caps Lock) i dlatego, jeśli strona klawiatury nie zostałaby podciągnięta do zasilania, oprócz komputera, również klawiatura mogłaby odczytać transmisję jako rozkaz dla niej. W takim przypadku nasz rozkaz zostałby zakłócony przez klawiaturę i z transmisji byłoby nici.

Teoretycznie wiemy już jak układ powinien działać. Brakuje jeszcze tylko programu obsługującego mikroprocesor. W artykule znajdziecie jego listing w języku BASCOM. Program ten z powodzeniem obsługiwał mój model, więc prawdopodobnie będzie działać i u Was. Ze względu na ograniczenia dotyczące artykułu, obcięte zostały komentarze. Podobnie jak w przypadku "Maszyny do pisania", zachęcam jednak do napisania programu samemu i traktowanie mojego listingu jedynie jako działający punkt odniesienia. Jeśli jednak ktoś nie ma ochoty tracić czasu nawet na przepisywanie, z pomocą przychodzi Internet. Ci, którzy mają do niego dostęp, mogą zassać program oraz zawartości pamięci szeregowych do sterowania różnymi programami spod adresu www.edw.com.pl lub <http://www.bloknet.pl/~rufus/pilot.zip>, a ci, którzy dobrodziejstwa Internetu nie znaleźli, zadowolili się muszą napisaniem programu samemu lub jego bezmyślnym (?) przepisanie, ale w końcu pisanie w BASCOM-ie to przyjemność!

Program mikroprocesorowy:

Scrystal = 11059200

Dim Nowy As Bit : Dim E As Bit : Dim P As Bit
Dim Com As Byte : Dim Adr As Byte
Dim K As Byte : Dim X As Byte : Dim A As Byte

Ziel Alias P1.0 : Czerw Alias P1.1 : Dta Alias P1.2
Clk Alias P1.3 : Blk Alias P3.5

Config Sda = P3.0 : Config Scl = P3.1
Config I2cdelay = 1

Reset Tcon.0 : On Int0 Rc5
Enable Int0 : Enable Interrupts

```

Do
  Bitwait Nowy , Set
  Disable Int0 : Reset Nowy
  I2cstart : I2cwbyte 160 : I2cwbyte Com
  I2cstart : I2cwbyte 161 : I2crbyte K , Nack : I2cstop
  If K = &HFF Then
    Com = Com + 64
    I2cstart : I2cwbyte 160 : I2cwbyte Com
    I2cstart : I2cwbyte 161 : I2crbyte K , Nack : I2cstop
    If K <> &HFF Then : Set E
    Else : Reset Czerw : Waitms 240 : Set Czerw
  End If
End If
If K <> &HFF Then
  While Clk = 0 : Waitms 2 : Wend : Reset Blk
  If E = 1 Then
    Reset E
    X = &HE0 : Gosub Przeslij
    X = K : Gosub Przeslij
    X = &HE0 : Gosub Przeslij
    X = &HF0 : Gosub Przeslij
    X = K : Gosub Przeslij
  Else
    X = K : Gosub Przeslij
    X = &HF0 : Gosub Przeslij
    X = K : Gosub Przeslij
  End If
  Set Blk : Reset Ziel : Waitms 240 : Set Ziel
End If
Enable Int0
Loop

```

```

Przeslij:
  A = 0
  If X.0 = 1 Then : A = A + 1 : End If
  If X.1 = 1 Then : A = A + 1 : End If
  If X.2 = 1 Then : A = A + 1 : End If
  If X.3 = 1 Then : A = A + 1 : End If
  If X.4 = 1 Then : A = A + 1 : End If
  If X.5 = 1 Then : A = A + 1 : End If
  If X.6 = 1 Then : A = A + 1 : End If
  If X.7 = 1 Then : A = A + 1 : End If
  A = A Mod 2
  If A = 0 Then : Set P : Else : Reset P : End If
  Dta = 0 : Delay : Clk = 0 : Delay
  Dta = X.0 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.1 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.2 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.3 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.4 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.5 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.6 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = X.7 : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = P : Clk = 1 : Delay : Clk = 0 : Delay
  Dta = 1 : Clk = 1 : Delay : Clk = 0 : Delay : Clk = 1
Return

```

```

Rc5:
  Getrc5(adr , Com)
  If Adr = 0 Then : Set Nowy
Else
  Reset Nowy
  Reset Czerw : Waitms 240 : Set Czerw
End If
Return

```

Montaż i uruchomienie

Układ jest bardzo prosty, więc montaż nie wymaga dużej uwagi. Należy zwrócić uwagę jedynie na to, aby kable łączące urządzenie z klawiaturą i komputerem nie były zbyt długie. Do obudowania urządzenia nadaje się, po małej obróbce, obudowa Z6. Przednią ściankę zastąpić można prostokątem z ciemnego, półprzezroczystego pleksi.

Ponieważ nasz układ udaje klawiaturę, nie będą potrzebne żadne sterowniki, aby go uruchomić. Jest to główna różnica pomiędzy tym urządzeniem a jego fabrycznymi odpowiednikami, które komunikują się

z komputerem przez port szeregowy. Dzięki temu nie będziemy musieli walczyć z niekompatybilnym systemem, który nigdy nie chce współpracować. Jedyne co trzeba zrobić to połączyć układ z komputerem i klawiaturą. Będzie działać pod każdym systemem i zawsze. To się dopiero nazywa "Plug and Play"!

Rafał Baranowski

e-mail: Rafal.Baranowski@bloknet.pl



Wykaz elementów

Rezystory:

R1,R2	560Ω
R3-R6	22kΩ
R7,R8	1kΩ
R9,R10	3,3kΩ
R11	220Ω

Kondensatory:

C1	100nF
C2,C6	100μF/16V
C3	10μF/16V
C4,C5	33pF

Półprzewodniki:

U1	AT89C2051
U2	AT24C01
U3	TFMS5360
T1-T4	BC557
D1	LED zielona
D2	LED czerwona

Pozostałe:

X1	rezonator ceramiczny 11,0592MHz
W1	gniazdo 5 pin DIN lub PS/2
W2	wtyk 5 pin DIN lub PS/2