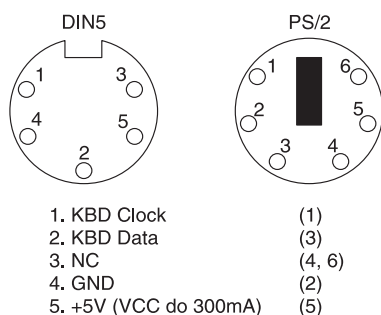


# Podstawy projektowania systemów mikroprocesorowych, część 9



Na rys. 30 przedstawiono wprowadzenia standardowych złącz klawiatur PC. Najlepiej stosować nieprzerobione w żaden sposób złącza (dostosować własne urządzenie do standardowego wtyku i rozmieszczenia sygnałów), choć w razie konieczności można zastosować dowolne inne połączenia - traci się jednak na kompatybilności ze standardem, co może utrudnić serwis w razie awarii. Na rys. 31 przedstawiono przykładowy sposób dołączenia klawiatury do mikrokontrolera. Zastosowano tu połączenie bezpośrednio z dodatkowymi rezystorami podciągającymi, dającymi lepszą odporność transmisji na zakłócenia. Klawiaturę należy zasilac napięciem +5 V, najlepiej tym samym, które zasilają współpracujący z nią mikrokontroler. Podczas projektowania układu zasilania należy uwzględnić dodatkowy pobór prądu, który - w zależności od modelu klawiatury - może dochodzić do 300 mA.

Jeżeli chodzi o sposób komunikacji klawiatury z otoczeniem, to główną jej cechą jest to, że w zależności



Rys. 30

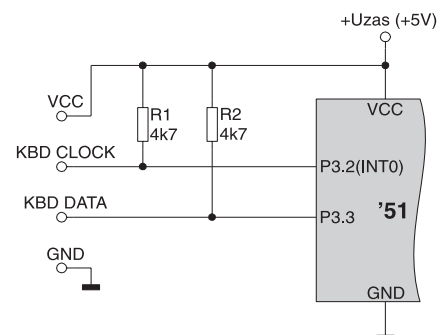
Zgodnie z zapowiedzią sprzed miesiąca, w tej części kursu przedstawiamy sposób wymiany danych pomiędzy mikrokontrolerem i standardową klawiaturą PC-AT. Zastosowanie takiej klawiatury to obecnie jeden z najtańszych sposobów wyposażenia systemu mikroprocesorowego w interfejs umożliwiający wprowadzanie danych.

od kierunku transmisji (z czy do klawiatury) przebiega ona w nieco inny sposób, choć zawsze jest to transmisja synchroniczna w takt sygnału zegarowego generowanego na linii *KBD Clock* przez klawiaturę. Proszym przypadkiem jest transmisja danych z klawiatury do mikroprocesora (przebiegi na rys. 32): następuje tutaj przesłanie słowa składającego się z 11 bitów: bitu startu (zawsze „0”), ośmiu bitów danych (począwszy od najmniej znaczącego), następnie bitu parzystości (typu *Odd* - dopełnienie do parzystej liczby jedynek w transmitowanym bajcie danych) oraz bit stopu, zawsze równy „1”. Zboczem taktującym jest opadające zbocze sygnału zegarowego. Prędkość transmisji nie jest zbyt duża i nie podlega żadnej dokładnej standaryzacji - częstotliwość przebiegu zegarowego zazwyczaj zawiera się w przedziale 20 do 30 kHz. Współpracujący mikrokontroler może sterować transmisją z klawiatury przez wymuszanie stanu niskiego na linii *KBD Clock* - stan niski oznacza blokadę transmisji i klawiatura rozpocznie transmitowanie ewentualnej danej dopiero po pojawieniu się na tej linii stanu wysokiego (podczas braku aktywności obydwie linie: *KBD Clock* i *KBD Data* znajdują się w stanie wysokim).

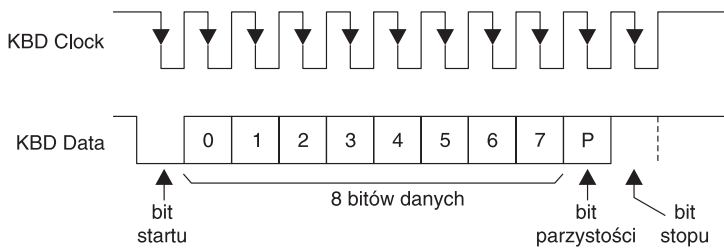
Nieco bardziej skomplikowana jest transmisja danych do klawiatury. Na rys. 33 przedstawiono przebiegi czasowe sygnałów podczas takiej transmisji. Inicjacja transmisji do klawiatury następuje przez wyzerowanie linii *KBD Clock* przez mikrokontroler (dokładnie chodzi o zbocze opadające na tej linii). W czasie trwania stanu niskiego generowanego na linii *Clock* (powinno to trwać nie krócej niż 60 ms, co gwarantuje potraktowanie tego

sygnału jako blokadę transmisji - klawiatura nie zacznie transmitować do mikrokontrolera w tym samym czasie) mikrokontroler zeruje linię *KBD Data*, a następnie ustawia stan wysoki na *KBD Clock*, co jest sygnałem rozpoczęcia transmisji do klawiatury. Wtedy kontrolę nad linią *KBD Clock* przejmuje klawiatura (w czasie do 10 ms) i rozpoczyna generację sygnału zegarowego. Transmisja danych przebiega podobnie jak wcześniej, z tym że linię *KBD Data* kontroluje mikrokontroler i to on wystawia kolejno bity przeznaczone do transmisji oraz bit parzystości. Po przesłaniu bitu parzystości mikrokontroler zwalnia linię *KBD Data* (stan wysoki), co jest interpretowane jako bit stopu - jeśli tego nie zrobi, to sygnał na linii *KBD Clock* będzie generowany tak długo, jak długo *KBD Data* będzie pozostawał w stanie niskim. Po przesłaniu bitu stopu klawiatura generuje dodatkowy takt sygnału zegarowego połączony z wyzerowaniem linii *KBD Data*, potwierdzając w ten sposób zakończenie transmisji.

Dokładny opis rozkazów sterujących klawiaturą oraz sposoby wykorzystania wszystkich jej funkcji Czytelnik znajdzie w literaturze poświęconej komputerom PC (np. w książce Piotra Metzgera „Anatomia PC”).



Rys. 31



Rys. 32

Z punktu widzenia konstruktora systemów mikroprocesorowych warto natomiast wspomnieć o podstawowych cechach użytkowych klawiatury PC. Jeżeli chodzi o transmisję z klawiatury do mikrokontrolera z nią współpracującego (najczęściej wykorzystywany przypadek), to następuje ona po każdym zdarzeniu związanym ze zmianą stanu klawiszy.

Klawiatura przesyła 8-bitowy kod naciśniętego klawisza w momencie jego wciśnięcia oraz ten sam kod powiększony o liczbę 128 (80H) w momencie puszczenia tego klawisza. Istnieją także grupy klawiszy, których naciskanie czy zwalnianie powoduje przesłanie większej liczby bajtów informacji (np. klawisze funkcyjne F1...F12 i inne) - dokładny ich opis Czytelnik znajdzie we wspomnianej literaturze. Jeżeli chodzi o transmisję danych do klawiatury (nie zawsze istnieje potrzeba jej wykorzystywania), to mikrokontroler sterujący przesyła klawiaturze rozkazy sterujące jej pracą. Jest możliwe m.in. sterowanie diodami świecącymi na klawiaturze, włączanie i wyłączanie klawiatury, zmiana prędkości autorepetycji oraz różne funkcje kontrolne.

Najczęściej chodzi nam o możliwość odczytywania stanu klawiatury (naciskanie klawiszy) - umożliwi nam to poniższy program (odpowiadający połączeniu klawiatury i mikrokontrolera według rys. 31) - list. 12.

Przedstawiony program w bardzo prosty sposób odczytuje kolejne bity pojawiające się na linii P3.3 w takt sygnału zegarowego doprowadzonego do linii P3.2. Został on tak napisany, aby łatwo można było zauważyć, które części procedury odbierającej dane są wykonywane zgodnie z kolejnością odbieranych bitów.

Opadające zbocze sygnału zegarowego wywołuje procedurę obsługi przerwania odczytującą kolejno wszystkie bity transmitowane przez klawiaturę. Do zabezpieczenia programu przed „utknięciem” na odbiorze któregoś z bitów (wskutek zakłóceń lub odłączenia klawiatury) wykorzystano licznik T1 zliczający czas, w któ-

rym musi zmieścić się cała transmisja. Jeżeli czas ten zostanie przekroczony, to zostaje wyzerowany licznik odbieranych bitów i program oczekuje na rozpoczęcie kolejnej transmisji.

Procedura obsługi przerwania INT0 po zakończeniu transmisji ustawia bit *NOWY* oznajmiający odebranie nowego kodu, który jest do-

stępny w zmiennej *KOD*, a kontrolny bit parzystości jest zwracany w zmiennej *PAR* - najczęściej nie istnieje potrzeba sprawdzania poprawności (przekłamania zdarzają się bardzo rzadko), jeśli zaszłyby jednak taka potrzeba, to należy pamiętać, że bit parzystości transmitowany przez klawiaturę jest negacją bitu parzystości dostępnego w rejestrze PSW mikrokontrolera (po wpisaniu otrzymanej danej do akumulatora).

Przystosowanie programu do transmisji w stronę klawiatury jest stosunkowo proste: należy zadbać o wygenerowanie sygnałów inicjujących na liniach zegara i danych, a następnie zamiast odczytywać stan linii *KBD Data*, należy po każdym zboczku opadającym sygnału zegaro-

## List. 12

```

;KOD - zmienna bajtowa przechowująca odebrany kod klawisza
;PAR - zmienna bitowa przechowująca odebrany bit parzystości
;NR_BIT - zmienna bajtowa przechowująca numer odebranego bitu
;NOWY - zmienna bitowa przechowująca informację o odebraniu nowego kodu

INCJALIZACJA:
CLR NOWY           ;zerowanie zmiennych
CLR PAR
MOV NR_BIT,#0
SETB P3.2
SETB P3.3

SETB EA           ;zaprogramowanie układu przerwań
SETB ET1
SETB EX0
SETB IT0         ;przerwanie od opadającego zbocza na INT0

MOV TMOD,#010H  ;licznik 1 w trybie 1 (licznik 16-bitowy)
...
...             ;(program główny)
...

PRZERWANIE_INT0: ;procedura obsługi przerwania zewnętrznego z linii
;INT0 (P3.2)

PUSH ACC         ;zapamiętanie
PUSH PSW        ;rejestrów na stosie

MOV A,NR_BIT
JZ BIT_STARTU   ;jeżeli bit startu (nr 0), to zignoruj go

CJNE A,#1,NIE_1 ;odbiór pierwszego bitu
CLR A           ;zerowanie
MOV C,P3.3     ;odczytanie bitu
MOV ACC,0,C    ;i wpisanie na najmłodszą pozycję akumulatora
SJMP ZAPAMIETAJ

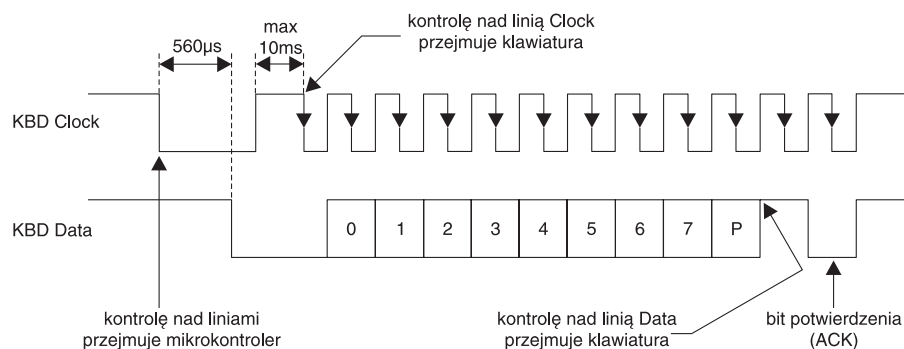
NIE_1:
CJNE A,#2,NIE_2 ;odbiór drugiego bitu
MOV A,KOD      ;odczytanie zapamiętanych już informacji
MOV C,P3.3     ;odczytanie bitu
MOV ACC,1,C    ;i wpisanie na odpowiednią pozycję akumulatora
SJMP ZAPAMIETAJ

NIE_2:
CJNE A,#3,NIE_3 ;odbiór trzeciego bitu
MOV A,KOD
MOV C,P3.3
MOV ACC,2,C
SJMP ZAPAMIETAJ

NIE_3:
CJNE A,#4,NIE_4 ;odbiór czwartego bitu
MOV A,KOD
MOV C,P3.3
MOV ACC,3,C
SJMP ZAPAMIETAJ

NIE_4:
CJNE A,#5,NIE_5 ;odbiór piątego bitu
MOV A,KOD
MOV C,P3.3
MOV ACC,4,C
SJMP ZAPAMIETAJ

```



Rys. 33

wego ustawiać na niej stan następnego bitu przeznaczonego do transmisji. Należy również zwiększyć o 1 liczbę możliwych stanów licznika bitów w związku z koniecznością

odebrania bitu potwierdzenia (nawet jeśli bit potwierdzenia będzie ignorowany przez program).

**Paweł Hadam, AVT**  
**pawel.hadam@ep.com.pl**

## List. 12 - cd.

```

NIE_5:
  CJNE A,#6,NIE_6 ;odbiór szóstego bitu
  MOV A,KOD
  MOV C,P3.3
  MOV ACC.5,C
  SJMP ZAPAMIETAJ

NIE_6:
  CJNE A,#7,NIE_7 ;odbiór siódmego bitu
  MOV A,KOD
  MOV C,P3.3
  MOV ACC.6,C
  SJMP ZAPAMIETAJ

NIE_7:
  CJNE A,#8,NIE_8 ;odbiór ósmego bitu
  MOV A,KOD
  MOV C,P3.3
  MOV ACC.7,C
  SJMP ZAPAMIETAJ

NIE_8:
  CJNE A,#9,NIE_9 ;odbiór bitu parzystości
  MOV C,P3.3 ;odczytanie bitu
  MOV PAR,C ;i zapamiętanie
  SJMP KONIEC

NIE_9:
  SETB NOWY ;odebranie bitu stopu - ustawienie
                ;informacji o nowym kodzie
  MOV NR_BIT,#0 ;wyzerowanie licznika bitów
  CLR TR1 ;zatrzymanie licznika
  SJMP KONIEC

ZAPAMIETAJ:
  MOV KOD,A ;zapamiętanie kodu
  INC NR_BIT ;zwiększenie licznika bitów
  SJMP KONIEC

BIT_STARTU:
  INC NR_BIT ;zwiększ nr bitu
  CLR NOWY ;zeruj bit (bo nadchodzi nowy kod i informacja
                ;o starym zostanie zamazana)
  MOV TH1,#0 ;włączenie licznika zabezpieczającego (timeout)
  MOV TL1,#0 ;dla kwarcu 12MHz daje to czas opóźnienia
  SETB TR1 ;około 65ms

KONIEC:
  POP PSW ;pobranie rejestrów
  POP ACC ;ze stosu
  RETI

PRZERWANIE_T1:
  ;obsługa przerwania od licznika T1
  CLR TR1 ;zatrzymanie licznika
  MOV NR_BIT,#0 ;wyzerowanie licznika bitów
  RETI

```