

Interfejs MIDI do komputerów Atari XE/XL

Prezentowany w artykule interfejs umożliwia rejestrację i odtwarzanie informacji w standardzie MIDI przez komputer typu Atari XE/XL. Urządzenie to stanowi zewnętrzną przystawkę pozwalającą, bez ingerencji do wnętrza komputera, podłączyć dowolne urządzenie MIDI (syntezator, keyboard lub moduł brzmieniowy).

Dzięki wyposażeniu przystawki w dwa wyjścia i jedno wejście, istnieje możliwość łączenia rozbudowanych zestawów MIDI (patrz przykłady na rys. 1). Urządzenie przetestowano z keyboardem firmy YAMAHA typu PSR-220, posiadającym 16-kanałowy moduł brzmieniowy.

Schemat elektryczny przystawki przedstawiono na rys. 2. Jak widać, jest on bardzo prosty i oparty na powszechnie dostępnych elementach. Spełnia wszys-

tkie wymagania dla standardu MIDI w zakresie poziomów sygnałów i galwanicznej izolacji urządzeń wchodzących w skład systemu. Transoptor IC2 izoluje galwanicznie urządzenie MIDI od komputera. Tranzystory T1 i T2 zapewniają dopasowanie sygnału do szyny szeregowej komputera.

Część nadawczą stanowi układ IC1. Bramka IC1c pracuje jako inwerter sygnału wysyłanego. Bramka IC1d pracuje jako przełącznik wyjść kluczując bramki IC1a i IC1b, które spełniają rolę nadajników prądowych o wydajności 5mA.

Dioda D2 sygnalizuje włączenie zasilania interfejsu napięciem przeznaczonym normalnie do sterowania silnikiem magnetofonu. Przed podłączeniem układu do komputera należy bezwzględnie sprawdzić prawidłowość połączeń, a także pobór prądu, który powinien wynosić ok. 15mA. Niekiedy, przy pojawieniu się kłopotów w odbiorze danych, trzeba będzie dobrać rezystor R2 (w zakresie od 1 do 2kΩ). Do sterowania interfejsem przez komputer ATARI wykorzystano jego wewnętrzny układ POKEY, będący nadajnikiem i odbiornikiem

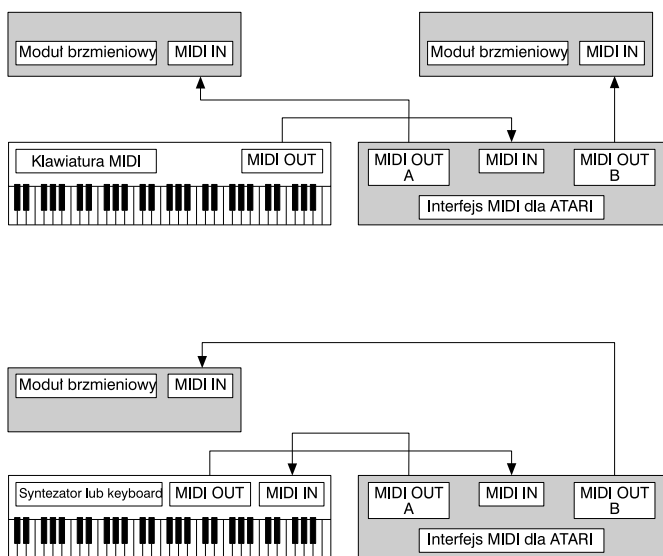


szeregowym z programowaną szybkością transmisji. Układ ten wyposażony jest w 4 generatory bazowe umożliwiające nastawienie jednej z następujących częstotliwości: 15kHz, 64kHz (dokładnie 63920Hz) lub 1,79MHz (1789760Hz) oraz jednobajtowe dzielniki częstotliwości (z możliwością łączenia w dwubajtowe). Z uwagi na wymaganą przez standard MIDI prędkość transmisji 31250 bitów/s, wykorzystać można tylko dwie wyższe szybkości. Do wyboru częstotliwości bazowej i łączenia dzielników jednobajtowych w podwójne służy rejestr *AUDCTL* (o adresie D208h). Jego bity posiadają następujące znaczenie:

bit 0 - wybór generatora bazowego (0=64 kHz, 1=15 kHz);

bit 1 - włącza filtr w generatorze 2 sterowany generatorem 4 (1=włączony);

bit 2 - włącza filtr w generatorze 1 sterowany generatorem 3 (1=włączony);



Rys. 1.

Listing 1.

```

*****
* TESTMIDI.ASM
*
* Odczyt danych MIDI
*
* QuickAssembler 1.0
*****

OptDOS EQU $100101
OptRAM EQU $010111

OPT OptRAM ;lub OptDOS dla kompilacji na dysk

Ekran EQU $58 ;Wektor z adresem ekranu
ClrScr EQU $P420 ;Adres procedury czyszczenia ekranu
AUDF1 EQU $D200 ;Cztery rejestry generatorow
AUDF2 EQU $D202 ;dla ukkladu POKEY
AUDF3 EQU $D204 ;
AUDF4 EQU $D206 ;
AUDCt1 EQU $D208 ;Rejestr kontroli generatorow POKEY
KBCode EQU $D209 ;Rejestr ostatnia naciśnietego klawisza
SKSRes EQU $D20A ;Reset statusu zlacza szeregowego
SerIn EQU $D20B ;Rejestr wejsciowy szeregowy
IRQEn EQU $D20E ;Zezwolenia przerwan IRQ
IRQSt EQU $D20F ;Status przerwan IRQ
SKCt1 EQU $D20F ;Rejestr kontroli zlacza szeregowego
SKStat EQU $D20F ;Rejestr statusu zlacza szeregowego
PAct1 EQU $D302 ;Rejestr kontroli portu A
NMIEn EQU $D40E ;Rejestr zezwolen przerwan NMI
IRQEnS EQU $10 ;cien IRQEn
RTClok EQU $14 ;zegar
SKCt1S EQU $232 ;rejestr cien SKCt1

ORG $9000

Start JSR ClrScr
      LDA RTClok
      CMP RTClok ;czekaj na NMI
      BEQ *-2 ;dla wyczyszczenia ekranu

      JSR MIDIOIn
      LDA KBCode
Wait   CMP KBCode ;czekaj na dowolny klawisz
      BEQ Wait
      JSR MIDIOIn

Quit   EQU * ;Program dla DOS'u konczy JMP ($A)
      RTS ;a w QuickAssemblerze RTS
      RTS
      RTS

* Procedura MIDIOIn zaprogramowanie
* interfejsu

MidiOn LDA #$00000000
      STA NMIEn ;blokada przerwan NMI
      LDA #52
      STA PAct1 ;właczenie zasilania interfejsu
      LDA #01110000 ;programowanie generatorow POKEY

* 6b-zeg. baz. gen.1 =2.217MHz
* 5b-zeg. baz. gen.3 =2.217MHz
* 4b-lacz. dz. gen.1+2
* 3b-lacz. dz. gen.3+4

      STA AUDCt1 ;31250 bitow/sek
      LDA #21
      STA AUDF3
      STA AUDF1
      LDA #0
      STA AUDF4
      STA AUDF2
      LDA #$01110011 ;odczyt wg gen.4, zapis wg. gen.2
      STA SKCt1S
      STA SKCt1
      STA SKSRes ;reset statusu zlacza szeregowego

      JSR Nowa ;ustawienie adresu nowej procedury odczytu
              ;ze zlacza szeregowego

      LDA #11100000 ;zewz. na przerwan BREAK,klawiatury,
              ;i odczytu ze zlacza szeregowego

      STA IRQEnS
      STA IRQEn
      RTS

MIDIOf LDA #60 ;wylaczenie zasilania interfejsu
      STA PAct1
      LDA #11000000 ;blokada przerwan ze zlacza szeregowego
      STA IRQEnS
      STA IRQEn
      LDA #01000000
      STA NMIEn ;Właczenie przerwan NMI
      JSR Stara ;przywrocenie starego adresu procedury
              ;odczytu ze zlacza szeregowego

* Ustaw nowa obsluge przerwan
Nowa   LDA $020A ;$20A -wektor z adresem procedury odczytu
      STA OlGADR ;ze zlacza szeregowego
      LDA $020A+1
      STA OlGADR+1
      LDA +Odczyt
      STA $020A
      LDA +Odczyt
      STA $020A+1
      LDA #0
      STA Pozyc
      RTS

* Przywroc stara obsluge przerwan
Stara  LDA OlGADR
      STA $020A
      LDA OlGADR+1
      STA $020A+1
      STA $020A+1
      RTS

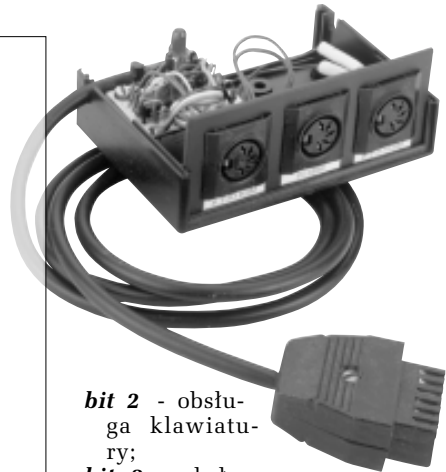
OldADR DTA a(0)
Pozyc DTA b(0)
TabHex DTA d'0123456789ABCDEF'

Odczyt LDA #11100000
      STA IRQEn ;zewzow na dalsze przerwan
      TAY
      PLA
      LDA SKStat
      STA SKSRes
      LDA SerIn ;wlasciwy odczyt danej
      CMP #5FE ;Ignoruj Active Sensing
      BEQ EndOdc
      PLA
      AND #11110000
      LSR @
      LSR @
      LSR @
      LSR @
      TAY
      LDA TabHex,Y
      LDY Pozyc
      STA (Ekran),Y ;wyswietlenie starszego polbajtu
      INC Pozyc
      PLA
      AND #00001111
      TAY
      LDA TabHex,Y
      LDY Pozyc
      STA (Ekran),Y ;wyswietlenie mlodszy polbajtu
      INC Pozyc
      LDA #128
      INY
      STA (Ekran),Y ;znak 'I' wskaze pozycie dla nowej
              ;danej

      EndOdc PLA ;zakonczenie procedury obslugi
              ;przerwan dla odczytu
      PLA
      RTI

      END ;usunac te linie przy kompilacji na dysk

      ORG Quit ;zakonczenie dla wersji pod DOS
      JMP ($A)
      ORG $2E0
      DTA a(Start)
      END
    
```



- bit 2** - obsługa klawiatury;
- bit 3** - obsługa przetworników A/C;
- bity 4 do 6** sterują szybkością transmisji:
- 000 - zewnętrznie;
- 001 - odczyt: wg generatora 4, zapis: zewnętrznie;
- 010 - wg generatora 4;
- 011 - kombinacja zabroniona;
- 100 - odczyt: zewnętrznie, zapis: wg generatora 4;
- 101 - kombinacja zabroniona;
- 110 - odczyt: wg generatora 4, zapis: wg generatora 2;
- 111 - jw., zablokowane wejście i wyjście taktujące;
- bit 7** - nadanie sygnału SPACE.

bit 3 - łączenie dzielników 3 i 4 w 16-bitowy (1=właczona);

bit 4 - łączenie dzielników 1 i 2 w 16-bitowy (1=właczona);

bit 5 - wybór zegara bazowego dla generatora 3 (0=wg bitu 0, 1=1,79 MHz);

bit 6 - wybór zegara bazowego dla generatora 1 (0=wg bitu 0, 1=1,79 MHz);

bit 7 - przełączanie rejestru przesuwającego.

Bity 1, 2, 7 w niniejszym zastosowaniu należy wyzerować. Rejestry dzielników częstotliwości znajdują się pod następującymi adresami:

- AUDF1=D200h - dzielnik generatora 1;
- AUDF2=D202h - dzielnik generatora 2;
- AUDF3=D204h - dzielnik generatora 3;
- AUDF4=D206h - dzielnik generatora 4.

Wymaganą prędkość transmisji uzyskuje się wpi-

sując do tych dzielników wartości określone z zależności

- w przypadku dzielników połączonych w 16-bitowy:

$$\text{Prędkość} = \frac{\text{częstość_bazowa}}{2} \cdot (\text{AUDF2} \cdot 256 + \text{AUDF1} + 7);$$

- w przypadku dzielników pojedynczych dla zegara 64kHz i 15 kHz:

$$\text{Prędkość} = \frac{\text{częstość_bazowa}}{2} \cdot (\text{AUDF2} + 1);$$

- w przypadku dzielników pojedynczych dla zegara 1,79 MHz:

$$\text{Prędkość} = \frac{\text{częstość_bazowa}}{2} \cdot (\text{AUDF2} + 4).$$

Identyczne wzory stosuje się dla dzielników AUDF3 i AUDF4.

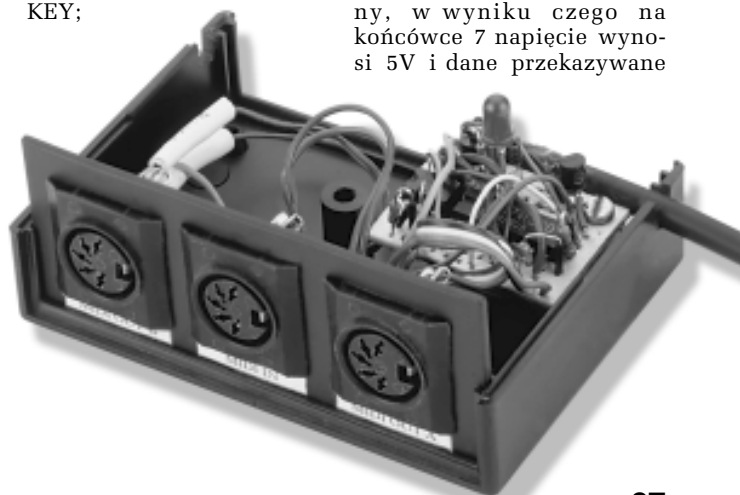
W przygotowanych programach testowych wykorzystano dzielniki połączone. Wpisując do AUDF1 i AUDF3 liczbę 21 oraz do AUDF2 i AUDF4 liczbę 0, uzyskano prędkość transmisji 31960 bitów/s. Jest to

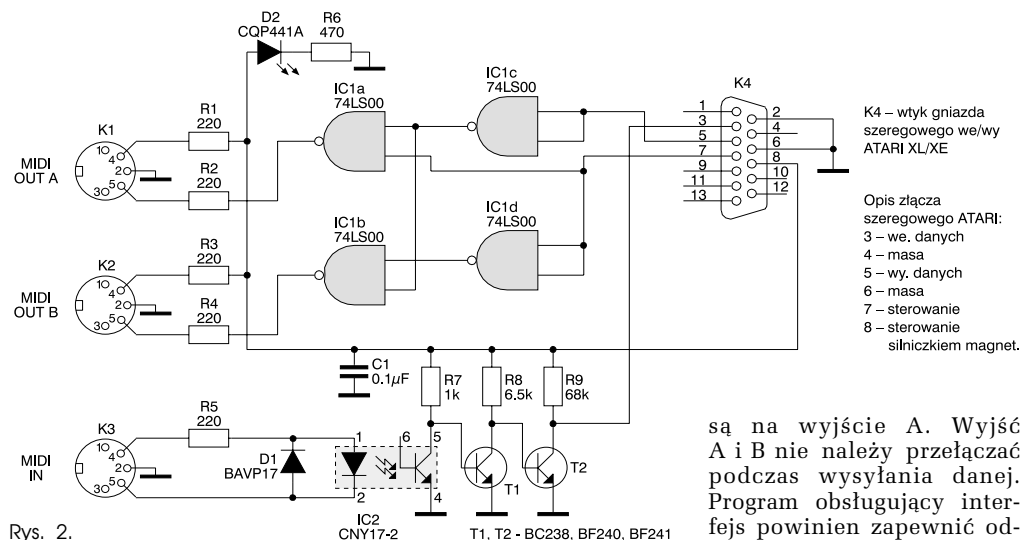
wartość nieco inna od wymaganej standardem, jednak jak stwierdzono w praktycznych testach, błąd poniżej 2% nie wpływa na jakość transmisji. Poprawne działanie uzyskano także wpisując liczbę 22 do AUDF1 i AUDF3 (30858 bitów/s).

Następnym ważnym rejestrem jest SKCt1 (o adresie D20Fh). Jego bity mają następujące znaczenie:

bit 0 i 1 - resetują POKEY;

Podczas prób okazało się, że poprawny odbiór uzyskuje się jedynie stosując blokadę wejścia i wyjścia taktującego (bity 4.6: 111). W czasie nadawania blokada ta nie ma znaczenia. Włączenie i wyłączenie urządzenia następuje przez zmianę bitu 3 rejestru PAct1 (adres D302h). Bit ten steruje napięciem na końcówce 8 szeregowego gniazda we/wy komputera (1=wyłączone, 0=właczona). W przypadku obsługi dwóch urządzeń MIDI istnieje możliwość przełączania między wyjście A i B. Steruje tym bit 3 rejestru PBCt1 (D303h). Normalnie jest on ustawiony, w wyniku czego na końcówce 7 napięcie wynosi 5V i dane przekazywane





Rys. 2.

Listing 2.

```

*****
* TESTMID2.ASM
*
* Wysłanie danych MIDI
* równoległe na wyjścia A i B
*
* QuickAssembler 1.0
*****

OptDOS EQU $100101
OptRAM EQU $010111

OPT OptRAM ;lub OptDOS przy kompilacji
na dysk

Adres EQU $60 ;Wskaznik dla danych MIDI
AUDF1 EQU $D200 ;Cztery rejestry generatorow
AUDF2 EQU $D202 ;dla układu POKEY
AUDF3 EQU $D204 ;
AUDF4 EQU $D206 ;
AUDC1 EQU $D208 ;Rejestr kontroli generatorow
POKEY
SKSRes EQU $D20A ;Reset statusu złącza
szeregowego
SerOut EQU $D20D ;Rejestr wyjsciowy szeregowy
SKCt1 EQU $D20F ;Rejestr kontroli złącza
szeregowego
SKScat EQU $D20P ;Rejestr statusu złącza
szeregowego
PAct1 EQU $D302 ;Rejestr kontroli portu A
PAct1 EQU $D303 ;Rejestr kontroli portu B
RTClock EQU $14 ;Zegar 1/50 sek.
SKC15 EQU $232 ;Cien SKCt1
WSync EQU $D40A ;Rejestr oczekiwania na
synchronizacje pozioma
Tempo EQU 10

ORG $9000

Start JSR MIDIOIn ;Ustaw wskaznik na poczatek
danych MIDI
LDA > Dane
STA Adres
LDA > Dane
STA Adres+1

Play LDA #0
LDA (Adres),Y ;Wczytaj dana MIDI
CMP #F8 ;Czy to koniec danych
BEQ EndDta
CMP #F8 ;Czy to F8-odczekanie
BNE OutDat ;Jesli NIE wyslij dana
MIDI JSR Wait ;Jesli TAK czekaj
JMP Nastep ;Odczyt nastepnej danej
OutDat LDX #53C ;Wlacz wyscie A
interfejsu
STX PAct1
STA SerOut ;Wyslij dana na wyscie A
STA WSync ;Czekaj min. 5 linii
ekranu
STA WSync ;tzn. na wyslanie 10
bitow
STA WSync ;z predkoscia 31250
bitow/sek
STA WSync
STA WSync
STA WSync
LDX #534 ;Wlacz wyscie B
interfejsu
STX PAct1
STA SerOut ;Wyslij dana na wyscie B
STA WSync ;i poczekaj min. 5 linii
ekranu
STA WSync
STA WSync
STA WSync
STA WSync
Nastep INC Adres ;Ustaw wskaznik na nowa
dana
BNE *+4
INC Adres+1
JMP Play
EndDta LDX #53C ;Odtworzenie typowej
wartosci
STX PAct1 ;dla PAct1
JSR MIDIOIn ;Wlacz zasilanie
interfejsu MIDI
;Koniec danych

Quit EQU * ;Program dla DOS'u konczy
JMP($A)
RTS ;a w QuickAssemblerze RTS
RTS
RTS
RTS

Wait LDY #Tempo ;Odczekanie Tempo*1/50 sec
Wait1 LDA RTClock ;Czekaj 1/50 sec
CMP RTClock ;Czekaj 1/50 sec

*****
* Dane MIDI
*
* P8h - czekaj
* F1h - koniec danych
* inne dane sa wyslane
*
*****
Dane DTA
b($F0),b($7E),b($7F),b(9),b(11),b($F7) ;GM Sys
On
DTA b($F8)
;Odstep
DTA b($90),b($59),b($53),b($F8)
;NotecOn,odstep
DTA b($80),b($59),b(0)
;NotecOf
DTA b($90),b($54),b($53),b($F8)
;itd.
DTA b($80),b($54),b(0)
DTA b($90),b($59),b($53),b($F8)
DTA b($80),b($59),b(0)
DTA b($90),b($62),b($53),b($F8)
DTA b($80),b($62),b(0)
DTA b($90),b($61),b($53),b($F8)
DTA b($80),b($61),b(0)
DTA b($90),b($54),b($53),b($F8)
DTA b($80),b($54),b(0)
DTA b($90),b($61),b($53),b($F8)
DTA b($80),b($61),b(0)
DTA b($90),b($64),b($53),b($F8)
DTA b($80),b($64),b(0)
DTA b($90),b($62),b($53),a($F8F8)
DTA b($80),b($62),b(0)
DTA b($90),b($66),b($53),a($F8F8)
DTA b($80),b($66),b(0)
DTA b($90),b($66),b($53),b($F8)
DTA b($80),b($66),b(0)
DTA b($90),b($58),b($53),a($F8F8)
DTA b($80),b($58),b(0)
DTA b($90),b($66),b($53),a($F8F8)
DTA b($80),b($66),b(0)
DTA b($90),b($59),b($53),b($F8)
DTA b($80),b($59),b(0)
DTA b($90),b($54),b($53),b($F8)
DTA b($80),b($54),b(0)
DTA b($90),b($62),b($53),b($F8)
DTA b($80),b($62),b(0)
DTA b($90),b($61),b($53),b($F8)
DTA b($80),b($61),b(0)
DTA b($90),b($64),b($53),b($F8)
DTA b($80),b($64),b(0)
DTA b($90),b($62),b($53),a($F8F8)
DTA b($80),b($62),b(0)
DTA b($90),b($59),b($53),a($F8F8),a($F8F8)
DTA b($80),b($59),b(0)
DTA b($F1)
;Koniec danych

END ;usunac te linie przy
kompilacji do pliku

ORG Quit ;zakonczenie dla wersji
plikowej pod DOS
JMP ($A)
ORG $220
DTA a(Start)
END
    
```

WYKAZ ELEMENTÓW

- Rezystory**
R1..R5: 220Ω
R6: 470Ω
R7: 1kΩ
R8: 6.5kΩ
R9: 68kΩ
- Kondensatory**
C1: 100nF
- Półprzewodniki**
T1, T2: BC547, BC237, BC238 (lub BF241, BF240 - zastosowano je w modelowym urządzeniu)
D1: BAVP17 lub podobna
D2: LED dowolna np. CQP441A
IC1: UCY74LS00
IC2: CNY17-2, CNY17-3
- Różne**
K1, K2, K3: gniazda DIN 5-stykowe
K4: 13-stykowy wtyk złącza szeregowego dla komputera ATARI XL/XE

są na wyjście A. Wyjść A i B nie należy przełączać podczas wysyłania danej. Program obsługujący interfejs powinien zapewnić odbiór danych MIDI za pośrednictwem przerwań maskowalnych IRQ. Zapewnia to optymalne wykorzystanie niewielkiej mocy obliczeniowej mikroprocesora 6502. Ważne jest, by na czas odbioru komunikatów MIDI wyłączyć obsługę przerwań niemaskowalnych NMI (wpisując zero do rejestru NMIEn o adresie D40Eh). W przeciwnym razie nastąpią opóźnienia w obsłudze przerwań objawiające się gubieniem pojedynczych bajtów w czasie odbioru danych. Wysyłanie sekwencji MIDI jest znacznie prostsze, nie wymaga bowiem ingerencji w przerwanie IRQ i NMI. Po zaprogramowaniu prędkości transmisji wpis dowolnej danej do rejestru SerOut (adres D20Dh) powoduje wysłanie jej przez złącze szeregowo. Przed wpisem kolejnej danej należy odczekać min. 1/3125 sekundy, tj. czas potrzebny na wysłanie 10 bitów (bit startu, 8 bitów danych i bit stopu). Najprościej uzyskać to przez 5-krotne wpisanie pod adres D40Ah (rejestr oczekiwania na synchronizację poziomą WSync) dowolnej danej. Spowoduje to odczekanie przez mikroprocesor wymaganego czasu. Efektywniejszą metodą byłoby wykorzystanie przerwań IRQ do poinformowania procesora o zakończeniu wysyłania danej. Z uwagi na obszerność problemu zainteresowanych odsyłam do literatury.

Dla ilustracji przedstawionych zasad programowania układu POKEY przedstawiono dwa listingi

programów w asemblerze mikroprocesora 6502. Pierwszy program TESTMID1.ASM (list. 1) pozwala na odbiór danych MIDI i wyświetlenie ich na ekranie monitora. Drugi TESTMID2.ASM (list. 2) odtwarza przykładową sekwencję MIDI jednocześnie na dwóch urządzeniach podłączonych do wyjść A i B w tempie określonym znacznikami F8h. Do obsługi przedstawionego interfejsu zostały opracowane również bardziej zaawansowane programy umożliwiające rejestrację i odtwarzanie sekwencji MIDI z możliwością zapisu na dyskietce lub kasecie a także odtwarzanie sekwencji z plików SMF (ang. Standard Midi Files). Programów tych nie zamieszczono z uwagi na ich obszerność.

Dla zainteresowanych problemem służę w miarę możliwości pomocą pod adresem e-mail: ixkuczek@kki.net.pl
Ireneusz Kuczek,
ixkuczek@kki.net.pl

Literatura:
[1] - W. Zientara, Mapa pamięci ATARI XL/XE procedury wejścia/wyjścia. SOETO W-wa 1988.
[2] - J. Ruszczyc, Asembler 6502. SOETO W-wa 1987.
[3] - Radioelektronik 12/88, Interfejs MIDI do komputera ZX Spectrum.