

Halina NOWAK

Piotr STRÓŻYNA

PROGRAMOWE MOŻLIWOŚCI PRZYSTOSOWANIA KOMPUTERA MERA-60
DO STEROWANIA INTERFEJSEM IEC-625

Streszczenie. W pracy przedstawiono problemy programowej kontroli komunikacji za pośrednictwem interfejsu IEC-625 (IEEE-488) w systemach pomiarowych z komputerem pełniącym funkcję jedyne go kontrolera. Zaprezentowano trzy przykładowe rozwiązania firm światowych, a następnie przedstawiono propozycje rozwiązań dla maszyny cyfrowej MERA-60. Na zakończenie dokonano oceny przytoczonych rozwiązań pod kątem ich przydatności dla użytkownika.

1. Wstęp

Wprowadzony w połowie lat siedemdziesiątych standard IEC-625 (IEEE-488) zaprojektowano w firmie Hewlett-Packard w wyniku potrzeby stworzenia standardu, umożliwiającego połączenie produkowanych przez różne firmy przyrządów pomiarowych w system, przenoszący wyniki serii pomiarów bez ingerencji człowieka.

Standard IEC-625 jest standardem typu magistralowego, przeznaczonym dla programowanych urządzeń pomiarowych o bitowo-równoległym i bajtowo-szeregowym sposobie przesyłania informacji.

Zaletami jego są:

- asynchroniczny sposób transmisji (do 1 Mb/s), umożliwiający łączenie przyrządów o różnych szybkościach transmisji,
- możliwość przesyłania danych od urządzenia nadawczego do urządzeń odbiorczych bez udziału urządzenia kontrolującego,
- wygodne okablowanie, pozwalające na prostą rozbudowę systemu.

Dokładny opis standardu zawierają prace [1] i [2].

Konfiguracje systemów pomiarowych konstruowanych na podstawie standardu IEC można podzielić na:

- systemy bez kontrolera,
- systemy z 1 kontrolerem,
- systemy z wieloma kontrolerami.

System bez kontrolera posiada oczywiście najmniejsze możliwości, mianowicie dopuszczalny jest tylko jeden sposób transmisji - od jedyne go w systemie nadawcy do jednego lub kilku odbiorców. Urządzenia muszą być ręcznie ustawiane do nadawania lub odbioru.

Wprowadzenie kontrolera do systemu pozwala na pracę kilku nadawców w systemie i zwiększa zakres wykonywanych przez urządzenia operacji.

W przypadku większej liczby kontrolerów jeden z nich musi być zaprojektowany jako jednostka nadrzędna zarządzająca całym systemem.

W większości zastosowań wystarczającą i najwygodniejszą jest konfiguracja z jednym kontrolerem. Umożliwia ona realizację skomplikowanych zadań, równocześnie zaś jest prosta ze względu na brak problemów, związanych z przekazywaniem sterowania między kilkoma kontrolerami.

Obecnie w kraju coraz większą popularnością cieszy się komputer MERA-60, produkowany w kooperacji z ZSRR. Opracowano również układ łączący komputer z magistralą IEC-625, będący funkcjonalnym odpowiednikiem pakietu IBV-11 firmy DEC. Uwzględniając istniejącą w kraju produkcję urządzeń wyposażonych w interfejs IEC, za celowe należy uznać stworzenie oprogramowania, pozwalającego użytkownikom komputera MERA-60 na zarządzanie systemem, wykorzystującym ten standard.

W dalszej części artykułu omówiono przykłady środków programowych, oferowanych przez różne firmy światowe, jak też propozycje autorów przystosowania komputera MERA-60 do pełnienia funkcji kontrolera w systemie.

2. Przykładowe rozwiązania firmowe komunikacji z magistralą IEC-625 za pośrednictwem języków wyższego rzędu

2.1. Rozwiązanie firmy Hewlett Packard

Przytoczone rozwiązanie dotyczy systemu pomiarowego opartego na jednostce centralnej HP 9830A Calculator [4]. Językiem programowania Calculatora jest BASIC rozszerzony w stosunku do standardu o specjalną instrukcję współpracy z interfejsem IEC. Instrukcja ta (CMD) pozwala na:

- zadresowanie nadawcy i odbiorców,
- wysłanie danych dla zaprogramowania urządzeń pomiarowych.

Instrukcja ma następującą postać:

```
CMD <łańcuch-komend> [ , <łańcuch-programujący> ]
```

<łańcuch-komend> - służy do zadresowania nadawcy i odbiorców,

<łańcuch-programujący> - służy do przesłania danych programujących do przyrządów.

W niektórych wersjach systemu stała łańcuchowe (np. "1234ABCD") można zastąpić nazwami zmiennych łańcuchowych (np. A\$).

Poniżej zilustrowano przykładem wykorzystanie instrukcji CMD:

```
10 CMD "7U3", "FOR4T1M3E"
```


Poszczególne ciągi bitów odpowiadające znakom ASCII w instrukcji CMD mają następujące znaczenie:

- ? - komenda UNL - unieważniająca dotychczasowe zaadresowanie odbiorników,
- U - zaadresowanie nadajnika (HP 9830A Calculator),
- 3 - zaadresowanie odbiornika (woltomierz cyfrowy),
- FO - pomiar napięcia stałego,
- R4 - zakres do 10 V,
- T1 - natychmiastowe wewnętrzne wyzwalenie,
- M3 - tryb pracy z zaadresowanym pojedynczym wyjściem,
- E - inicjacja pomiaru.

Komunikaty interfejsowe i komunikaty urządzeń (dane) można również przesłać, korzystając ze standardowych instrukcji BASIC-a - OUTPUT lub WRITE. Język BASIC dla HP 9830A posiada ponadto deklarację FORMAT pozwalającą na precyzyjne wyspecyfikowanie postaci przesyłanych danych. Poniższe 3 instrukcje powodują wysłanie komendy REN (zezwolenie na zdalne sterowanie):

```
10 CMD "?U"
20 FORMAT B
30 OUTPUT (13,20) 786;
```

- "?U" - oznacza wysłanie komunikatu UNL i zaadresowanie Calculatora do nadawania,
- B w deklaracji FORMAT oznacza wysłanie jednego bajtu,
- liczby w instrukcji OUTPUT oznaczają:
 - 13 - kod selekcyjny magistrali IEC,
 - 20 - numer linii deklaracji FORMAT,
 - 786 - kod komendy REN.

Odbiór danych osiąga się wykorzystując standardową instrukcję wejścia ENTER w połączeniu z instrukcjami FORMAT i CMD.

Instrukcja CMD pozwala ponadto na wysłanie komend uniwersalnych (LLO, DCL itd.), adresowych (np. GTL) i bezadresowych (UNL, UNT).

Poza tym opisywana wersja BASIC-a umożliwi kontrolę szeregową i obsługę żądań. Funkcje BASIC-a STAT i RBYTE pozwalają na sprawdzenie, czy wystąpiło żądanie obsługi i na odczyt statusu magistrali.

Rozszerzona wersja BASIC-a, którą stosuje Hewlett-Packard, jest bardzo wygodna dla użytkownika. Praktycznie, rozszerzenie polega na wprowadzeniu jednej instrukcji CMD komunikacji z interfejsem IEC. Wadą tego rozwiązania jest konieczność programowego sprawdzania, czy wystąpiło żądanie obsługi, co uniemożliwia korzystanie z mechanizmu przerwania SRQ.

Na powyższym rozwiązaniu oparto zalecenia RWPG dotyczące programowej kontroli komunikacji za pośrednictwem interfejsu IEC-625.

2.2. Rozwiązanie firmy Philips

Firma Philips również opracowała rozszerzony BASIC do programowania kontrolera PM 4410 magistrali IEC [3]. Zbiór instrukcji BASIC-a stworzony do dyspozycji użytkownika interfejsu IEC jest bogatszy w stosunku do rozwiązania przedstawionego w punkcie poprzednim. Na zbiór ten składają się następujące instrukcje:

a) INIT - bezparametrowa instrukcja powodująca zerowanie interfejsu (komunikat IFC) i ustawienie urządzeń w stan zdalnej kontroli (komunikat REN).

b) PRINT - standardowa instrukcja BASIC-a służąca do przesyłania informacji z kontrolera do innego urządzenia, np.:

```
PRINT # 11,4: "ABCD1234".
```

Powyższa instrukcja adresuje urządzenie o adresie 11 do odbioru, powoduje wysłanie komunikatu SDC (o kodzie dziesiętnym 4), a następnie wysyła ciąg znaków "ABCD1234".

c) INPUT - standardowa instrukcja komplementarna do poprzedniej, np.

```
INPUT # 12,A$.
```

W tym przypadku kontroler odbiera od urządzenia o adresie 12 ciąg znaków i zapamiętuje go w zmiennej łańcuchowej o nazwie A\$.

d) CHEND - określenie ogranicznika danych przyjmowanych przez kontroler.

e) PWRITE - instrukcja służy do adresowania urządzeń oraz wysyłania komunikatów interfejsowych i danych do przyrządów, np.:

```
PWRITE 63,64,35,36: "ABCD1234ETX".
```

Powyższa instrukcja wysyła komunikat UNL (o kodzie 63), adresuje urządzenia:

- do nadawania o adresie 64 (kontroler),

- do odbioru o adresach 35 i 36,

a następnie wysyła do odbiorców ciąg znaków "ABCD1234" zakończony znakiem ETX.

f) PREAD - służy do przyjęcia przez kontroler jednego bajtu informacji z magistrali.

g) AT SRQ <nr> - deklaracja numeru linii programowej (<nr>), od której rozpocznie się program po przerwaniu SRQ.

h) POLL - instrukcja kontroli szeregowej; pozwala na pobranie statusu urządzeń żądających obsługi, np.:

```
POLL 10: K,L.
```

Instrukcja ta powoduje przesłanie pod zmienną o nazwie L statusu urządzenia o adresie 10. Natomiast zmienna K przyjmuje wartość 1 względnie 0 zależnie od tego, czy urządzenie to żądało obsługi czy nie.

Powyższy przegląd instrukcji wskazuje, że programowe możliwości kontroli pracy magistrali IEC są większe i bardziej elastyczne w porównaniu z możliwościami BASIC-a firmy Hewlett-Packard. W szczególności wykonywany program może być przerwany żądaniem obsługi SRQ przez urządzenie pomiarowe.

wa. Z drugiej jednak strony brak deklaracji FORMAT wymaga dodatkowej konwersji danych przyjmowanych przez komputer.

Należy zaznaczyć, że w obu opisanych przypadkach w skład oprogramowania podstawowego systemów wchodziły procedury sterujące kartami sprzężenia interfejsu IEC-625 z jednostką centralną (zwane powszechnie driverami lub handlerami).

2.3. Rozwiązanie firmy Digital Equipment Corporation

Rozwiązanie to dotyczy systemu operacyjnego RT-11 na maszynie PDP-11. Firma wyposażała system w handler interfejsu IEC i bibliotekę podprogramów (o nazwie IBLIB), ułatwiających komunikację z magistralą. Podprogramy te najkorzystniej jest - ze względu na bardzo proste przekazywanie parametrów - wywoływać z programów napisanych w FORTRAN-ie. Jednakże możliwa jest również współpraca z procedurami biblioteki IBLIB na poziomie innych języków, np. assemblera. Dołączanie podprogramów odbywa się na etapie konsolidacji zbiorów typu OBJECT.

Jako przykład zaprezentowano poniżej dwie możliwe formy wywołania procedury IBRECV służącej do odbioru komunikatów.

a) IBRECV może być z programu napisanego w FORTRAN-ie wywoływane jako podprogram:

```
CALL IBRECV (<tablica>, <limit> [, [<nadajnik>][, <odbiornik>]])
```

Znaczenie parametrów jest następujące:

<tablica> - nazwa tablicy, w której zapamiętywane są odebrane dane,

<limit> - maksymalna liczba zapamiętanych w tablicy danych,

<nadajnik> - opcjonalna specyfikacja nadawcy,

<odbiornik> - opcjonalna specyfikacja odbiorcy.

b) Procedura ta może być również wywołana jako funkcja:

```
D = IBRECV (<tablica>, <limit> [, [<nadajnik>][, <odbiornik>]])
```

W tym przypadku zmienna D przyjmuje wartość równą liczbie znaków przyjętych przez odbiornik (musi być spełniona nierówność $D \leq \langle \text{limit} \rangle$).

Poszczególne procedury biblioteki IBLIB umożliwiają nadawanie, odbiór i przesyłanie (bez zapamiętywania w komputerze) informacji zarówno przy odblokowanych, jak i zablokowanych przerwaniami. Podprogramy biblioteczne pozwalają również na:

- zablokowanie i odblokowanie zdalnego sterowania (bezparametrowe procedury IBRDA i IBREN),
- zerowanie interfejsu (bezparametrowa procedura IBIFC),
- wysyłanie komend uniwersalnych i adresowych (procedury IBLLO, IBSDC, IBDCI, IBUNT, IBGTL itd.),
- kontrolę szeregową oraz równoległą, odczyt statusu przyrządu,

- testowanie flagi żądania obsługi (SRQ),
- specyfikację procedury obsługi,
- ustalenie znacznika końca komunikatu,
- zadanie interwału czasowego oczekiwania na zakończenie transmisji.

Ponadto w bibliotece istnieje podprogram obsługi błędów, drukujący informacje o błędach powstałych w związku z komunikacją za pośrednictwem magistrali IEC.

Mnogość procedur biblioteki IBLIB pozwala programiście na pełne wykorzystanie możliwości standardu IEC-625, na tworzenie wyszukanych i oryginalnych programów użytkowych. Tych pełnych możliwości nie stwarzają rozszerzenia BASIC-a opisane w punktach 2.1 i 2.2. Jednakże korzystanie z biblioteki wymaga od programisty większego doświadczenia, a programy sterowania interfejsem IEC pisane w FORTRAN-ie współpracujące z opisanymi procedurami wymagają dłuższego czasu przygotowania i uruchomienia. Są przy tym narażone na większą ilość błędów w porównaniu z podobnymi programami napisanymi w BASIC-u.

Zaznaczyć należy, że czas realizacji programów w FORTRAN-ie jest zauważalnie krótszy od czasu wykonywania programów w BASIC-u. Wynika to z faktu, że na ogół translatorzy FORTRAN-u są kompilatorami, a BASIC-a interpreterami.

Na zakończenie trzeba wspomnieć, że opisane rozwiązanie firmy DEC nie jest jedynym dla systemu operacyjnego RT-11.

3. Propozycje rozwiązań dla komputera MERA-60

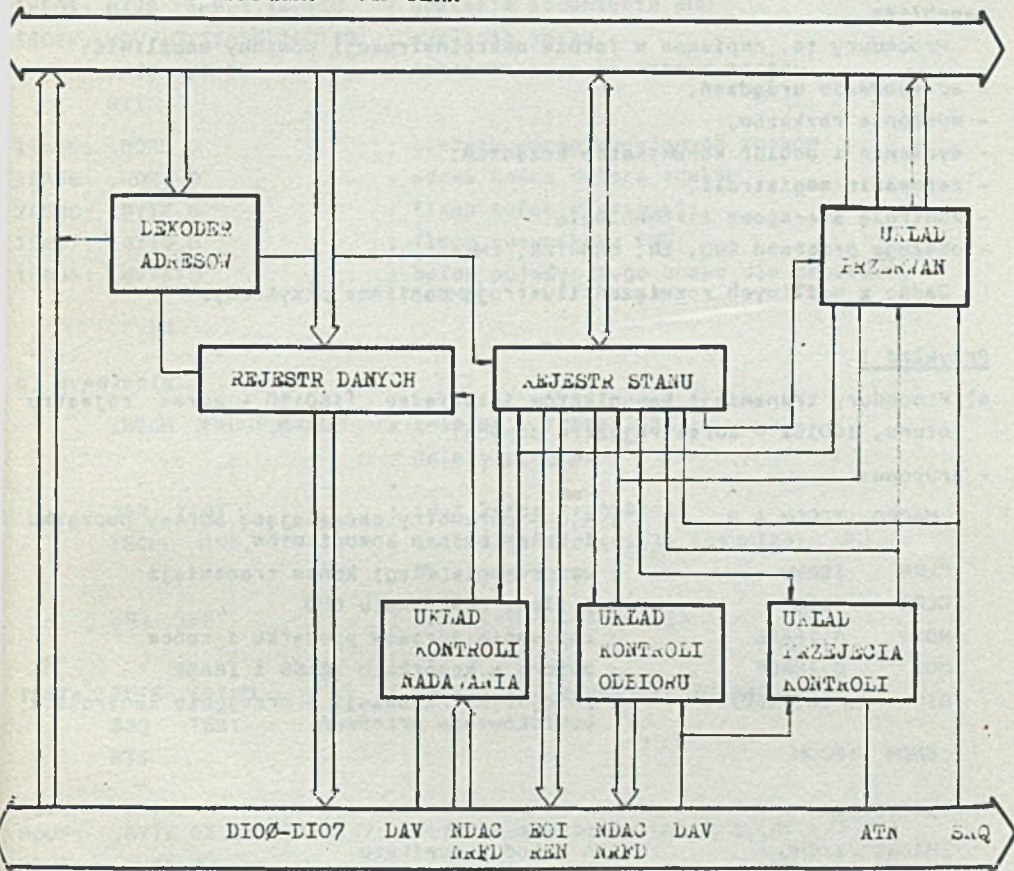
Poniższe rozwiązania dotyczą realizacji sprzężenia komputera z magistralą IEC-625 w sposób programowy, przy czym komputer przewidziany jest jako jedyny kontroler w systemie. Część urządzeniową stanowią bufony i układy niezbędne ze względu na uzależnienie czasowe. Przykładem takiego rozwiązania konstrukcyjnego jest pakiet IBV-11 firmy DEC, na podstawie którego zaprojektowano układ interfejsu dla MERA-60 (rys. 1, tabl. 1).

Tablica 1

Struktura rejestru stanu układu interfejsu

Nr bitu	Nazwa	Znaczenie
15	SRQ	żądanie obsługi urządzenia
14	ER2	próba wysłania bajtu przy NDAC = HIGH
13	ER1	konflikt linii IFC, ATN, REN
12, 11	-	niewykorzystane
10	CMD	gotów do wysłania komunikatu interfejsu
9	TKR	gotów do wysłania komunikatu urządzeń
8	LNR	gotów do wczytania bajtu

MAGISTRALA KOMPUTERA



MAGISTRALA IEC-625

Rys. 1. Schemat układu sprzężenia z magistralą IEC-625

cd. tablicy 1

Nr bitu	Nazwa	Znaczenie
7	ACC	określenie sposobu odbioru bajtu
6	IE	zezwolenie na przerwanie
5	TON	ustawienie do nadawania komunikatów urzędzeń
4	LON	ustawienia do odbioru
3, 2, 1	IBC, REM, EOP	ustawienie linii: IFC, REN, EOI
0	TCS	synchroniczne przejście kontroli

A. Komunikacja z poziomu assemblera

Najprostsze rozwiązanie stanowi zespół procedur, napisanych w makro-assemblerze, zebranych w bibliotekę i dostępnych dla użytkownika z poziomu assemblera.

Procedury te, napisane w formie makroinstrukcji powinny umożliwić:

- adresowanie urządzeń,
- wysłanie rozkazów,
- wysłanie i odbiór komunikatów urządzeń,
- zerowanie magistrali,
- kontrolę szeregową i równoległą,
- obsługę przerw SRQ, ER, CMD-TKR, LMR.

Jedno z możliwych rozwiązań ilustrują poniższe przykłady.

Przykład 1

a) Procedury transmisji komunikatów interfejsu (160150 - adres rejestru stanu, 160152 - adres rejestru danych)

- grupowo:

```
.MACRO IECOM A,B          ; A,B - parametry określające adresy początku
                           ; i końca bufora komunikatów
CLRB IEEND                ; wyzerowanie flagi końca transmisji
CLRB IEEOI                ; i flagi komunikatu END
MOV A,IEASB               ; zapisanie adresów początku i końca
MOV B,IEASE               ; bufora w komórkach IEASB i IEASE
BIS #101,160150           ; inicjacja transmisji - przejęcie kontroli i
                           ; odblokowanie przerw
.ENDM IECOM
```

- pojedynczo:

```
.MACRO IECMD A            ; A - kod komunikatu
CLRB IEEND                ; wyzerowanie flagi końca transmisji
CLRB IEEOI                ; i flagi komunikatu END
MOVB #IEBUF,IEASB         ; przygotowanie w komórce IEASB
MOVB #IEBUF,IEASE         ; adresu kodu komunikatu
MOV A,IEBUF
BIS #101,160150           ; inicjacja transmisji - przejęcie kontroli i
                           ; odblokowanie przerw
.ENDM IECMD
```

b) Obsługa przerw CMD-TKR, wspólna dla transmisji rozkazów, adresów i danych

```
IECRQ: CMP IEASB,IEASE    ; czy wysłano ostatni lub przedostatni znak z
                           ; bufora?
        BLT IECEX         ; skok, jeśli ostatni znak
        BEQ IEEOI        ; skok, jeśli wysłano przedostatni znak
```



```

MOV B #1,IEEND ; ustawienie flagi końca transmisji
BIC #43,160150 ; zablokowanie układów nadajnika pakietu
RTI
IEEOS: BIS B IEEOI,160150 ; wysłanie komunikatu END
IECEX: MOV @ IEASB,160152 ; wysłanie znaku
INCB IEASB ; inkrementacja bieżącego adresu
RTI
IEASB: .WORD 0 ; bieżący adres wysyłanych znaków
IEASE: .WORD 0 ; adres końca bufora znaków
IEEND: .BYTE 0 ; flaga końca transmisji
IEEOI: .BYTE 0 ; flaga komunikatu END
IEBUF: .BYTE 0 ; bufor pojedynczego znaku dla procedury ; IECMD
.EVEN

```

c) Wywołanie

```

IECOM #PBUF,#KBUF ; inicjacja transmisji
. ; dalszy program
.
JSR TEST ; test końca transmisji
IECMD #95. ; inicjacja transmisji komunikatu UNT
. ; dalszy program
JSR TEST ; test końca transmisji
.
.
TEST: TSTB IEEND ; test flagi końca transmisji
BEQ TEST
RTS
.
PBUF: .BYTE 63.,32.,33.,47. ; bufor komunikatów UNL,LA1,LA2,LA3,TA
KBUF: .BYTE 77.
.EVEN

```

Przykład 2

a) Procedury transmisji danych

- nadawania

```

.MACRO IEDAT A,B,C ; A,B - parametry określające adresy początku
; i końca bufora znaków
; C - powoduje wysłanie równocześnie z ostat-
; nim znakiem komunikatu END
CLRB IEEND ; wyzerowanie flagi końca transmisji
CLRB IEEOI ; i flagi komunikatu END
MOV A,IEASB ; zapisanie adresów początku i końca
MOV B,IEASE ; bufora w komórkach IEASB i IEASE

```



```

.IIF NB C, MOVB #2, IEEOI ; przygotowanie komunikatu END
BIS #140, 160150 ; inicjacja transmisji - ustawienie do na-
.ENDM IEDAT ; dawania i odblokowanie przerwań

- odbioru

.MACRO IEREC, A, B, C ; A, B - parametry określające adresy począt-
; ku i końca bufora dla danych
; C - kod znaku kończącego odbiór

CLRB IEEND ; wyzerowanie flagi końca transmisji
MOV A, IEARB ; zapisanie adresów początku i końca bufora
MOV B, IEARE ; w komórkach IEARB i IEARE
MOV C, IEPATT ; zapisanie kodu znaku kończącego odbiór w
; komórce IEPATT

BIS #320, 160150 ; inicjacja odbioru - ustawienie do odbioru
; i odblokowanie przerwań

.ENDM IEREC

```

b) Obeżuga przerwania LNR

```

IELRQ: MOVB 160150, IEARB ; wczytanie znaku do bufora
INCB IEARB ; inkrementacja adresu bieżącego
TST 160152 ; test komunikatu END
BMI IELEX ; skok, jeśli wczytano ostatni znak
CMPB IEPATT, 160152 ; test znaku kończącego
BEQ IELEX ; skok, jeśli wczytano znak kończącego
CMP IEARB, IEARE ; test zapelnienia bufora
BLE IELRTI ; skok, jeśli bufor nie jest pełny
CMPB IEPATT, IEARE ; czy ostatni znak był znakiem kończącym?
BEQ IELEX ; skok, jeśli tak
MOVB #1, IEEND ; ustawienie flagi

IELEX: CLR 160152 ; akceptacja znaku
BIC #220, 160150 ; zablokowanie odbiornika
INCB IEEND ; ustawianie flagi końca transmisji
; (1 - jeśli odebrano wszystkie znaki,
; 2 - jeśli bufor za mały)

RTI

IELRTI: CLR 160152 ; akceptacja znaku
RTI

IEARB: .WORD 0 ; bieżący adres odbieranych znaków
IEARE: .WORD 0 ; adres końca bufora

```


c) Wywołanie:

```

IEREC #PBUF,#KBUF, WZOR; inicjacja odbioru danych
      : ; dalszy program
TST:  CMPB #1,IEEND      ; test końca transmisji i kompletności ode-
      : ; branych danych
      BGT  TST           ; skok, jeśli transmisja niezakończona
      BEQ  DALEJ        ; skok, jeśli odebrano wszystkie dane
      .PRINT KOMUN     ; bufor zbyt mały
DALEJ: ; dalszy program
      IEDAT #PDAN,#K DAN-1 ; inicjacja wysłania danych z bufora
TEST:  TSTB IEEND      ; test końca transmisji
      BEQ  TEST
      :
PBUF:  .-.+20          ; bufor dla danych
KBUF:  .BYTE 0
WZOR:  .BYTE 3        ; kod znaku kończącego odbiór (ETX)
PDAN:  .ASCII/123.456/ ; bufor z danymi
KDAN:  .BYTE 0
      .EVEN
KOMUN: .ASCIZ/PRZEPEŁNIENIE BUFORA-ODEBRANY CIAG NIEKOMPLETNY/
      .EVEN

```

Przykład 3

a) Procedura kontroli równoległej

```

.MACRO IEPOLL,A ; A - parametr określający adres komórki,
                ; w której ma być zapisany komunikat PPR
BIC #100,160150 ; zablokowanie przerwań
BIS #1,160150   ; przejęcie kontroli
BIT #2000,160150 ; test przejęcia kontroli
BEQ -.6
BIS #3,160150   ; wysłanie komunikatu IDY
NOP
MOVB 160152,A ; zapisanie komunikatu PPR
BIC #3,160150 ; koniec komunikatu IDY
.ENDM IEPOLL

```

b) Wywołanie

```

IECOM #POCZ,#KON ; inicjacja transmisji komunikatów
TEST: TSTB IEEND ; test końca transmisji
      BEQ  TEST
      IEPOLL PPR ; kontrola równoległa
      :
      :
POCZ: .BYTE 33.,142.,34.,5. ; bufor komunikatów LA1,PPC,PPE1,LA2,PPC

```


KON: .BYTE 143. ; 1 PPE2
 PPR: .BYTE 0
 .EVEN

B. Komunikacja z poziomu języka wyższego rzędu

Przedstawione poprzednio rozwiązanie jest proste, elastyczne i zapewniające wystarczającą dla większości zastosowań szybkość transmisji (wg szacunkowych obliczeń dla kierunku wyjściowego - 20 kb/s, dla kierunku wejściowego - 10 kb/s). Jest jednak dość uciążliwe ze względu na żmudny proces konwersji danych w assemblerze.

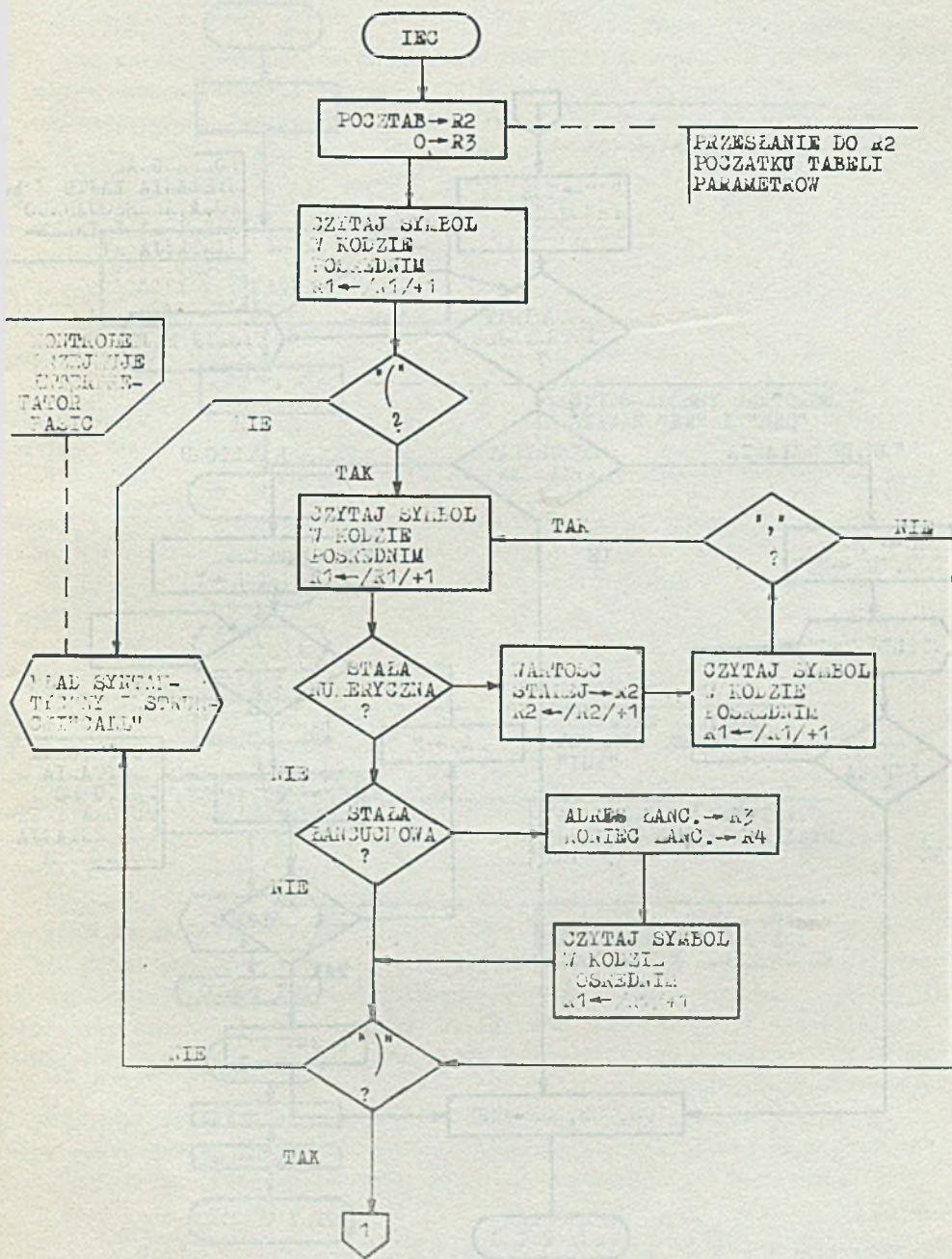
Korzystniejsza byłaby zatem komunikacja z magistralą IEC-625 z poziomu języka wyższego rzędu, np. BASIC. W tym celu należałoby rozszerzyć interpreter BASIC-60 o dodatkowe instrukcje, których postać byłaby np. dostosowana do zaleceń RWPG.

Inne rozwiązanie, nie wymagające ingerencji w strukturę interpretera polega na wykorzystaniu instrukcji "CALL" interpretera BASIC-60 umożliwiającej wykonanie programów, napisanych oddzielnie w assemblerze i dołączonych na etapie generacji interpretera. Postać instrukcji komunikacji z magistralą IEC może być następująca:

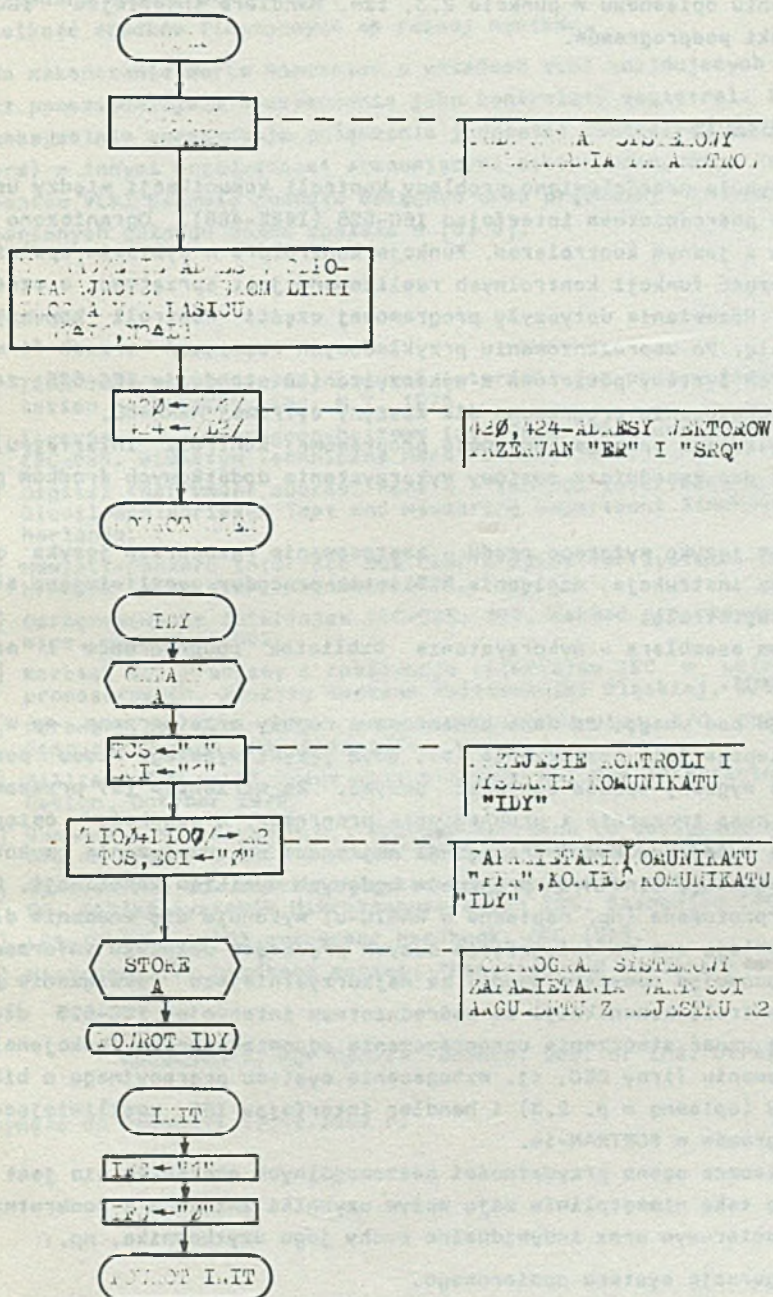
CALL "<nazwa>" [(<liczba parametrów>)]	- ogólna postać instrukcji
CALL "INIT"	- instrukcja zerowania magistrali
CALL "IDY" (<nazwa zmiennej>)	- instrukcja kontroli równoległej
CALL "RQER" (<nr linii 1>, <nr linii 2>)	- instrukcja ustalająca adresy programów obsługi przerw ER i SRQ
CALL "IEC" (<komunikat> [, (komunikat) ^N] [<nazwa zmiennej łańcuchowej>])	- instrukcja transmisji komunikatów

Jeśli przyjąć stałą liczbę parametrów instrukcji "CALL", wówczas można skorzystać z podprogramów systemu RT-60, sprawdzających poprawność instrukcji i wspomagających proces przekazywania parametrów między interpreterem a programem w assemblerze (rys. 2). Dla wygody użytkownika należałoby jednak założyć, że liczba parametrów instrukcji "CALL "IEC" jest zmienna - ze względu na zmienną liczbę wysyłanych rozkazów, adresów i danych. Zatem sprawdzenie poprawności i przekazanie parametrów muszą być realizowane przez program w assemblerze. Należy zaznaczyć, że powyższe rozwiązanie (zwłaszcza rozszerzenie interpretera o dodatkowe instrukcje), aczkolwiek bardzo wygodne dla użytkownika, są znacznie wolniejsze od przedstawionych w poprzedniej części procedur.

Kolejnym rozwiązaniem jest stworzenie translatora nowego języka, nastawionego na komunikację z magistralą IEC. Zestaw instrukcji tego języka powinien umożliwić sterowanie magistralą i przetwarzanie danych.



Rys. 2a. Schematy blokowe programów transmisji



Rys. 2c. Schematy blokowe programów transmisji cd.

Ze względu na kompatybilność oprogramowania komputerów MERA-60 i PDP-11, możliwe jest stworzenie oprogramowania odpowiadającego funkcjonalnie rozwiązaniu opisanemu w punkcie 2.3, tzn. handlera interfejsu IEC-62 i biblioteki podprogramów.

4. Zakończenie

W artykule przedstawiono problemy kontroli komunikacji między urządzeniami za pośrednictwem interfejsu IEC-625 (IEEE-488). Ograniczono się do systemów z jednym kontrolerem. Funkcje kontrolera w systemie spełnia komputer; część funkcji kontrolnych realizowana jest sprzętowo, a część programowo. Rozważania dotyczyły programowej części kontroli komunikacji z magistralą. Po zaprezentowaniu przykładowych rozwiązań trzech firm, produkujących systemy pomiarowe z wykorzystaniem standardu IEC-625, zaproponowano rozwiązania programowe dla maszyny cyfrowej MERA-60.

Przeglądając opisane koncepcje programowej kontroli interfejsu, można wyróżnić dwa zasadnicze poziomy wykorzystania dodatkowych środków programowych:

- a) poziom języka wyższego rzędu - zastosowanie rozszerzeń języka o specjalne instrukcje, względnie bibliotek procedur umożliwiające sterowanie magistralą,
- b) poziom assemblera - wykorzystanie bibliotek podprogramów i makroinstrukcji.

Biorąc pod uwagę, że dane pomiarowe z reguły przetwarzane są w procesorze, lepsze jest rozwiązanie (a), gdyż języki wyższego rzędu pozwalają w bardzo wygodny sposób operować danymi. Za wariantem (a) przemawia też krótszy czas tworzenia i uruchamiania programów. W aspekcie osiągnięcia możliwie dużej prędkości transmisji najlepsze są rozwiązania wykorzystujące realizację binarnych programów będących wynikiem kompilacji. Programy interpretowane (np. napisane w BASIC-u) wykonuje się znacznie dłużej i nie pozwalają one na osiągnięcie dużych prędkości przesyłu informacji.

Podsumowując powyższe uwagi, że najkorzystniejsze rozwiązanie programowej kontroli komunikacji za pośrednictwem interfejsu IEC-625 dla MERA-60 można uznać stworzenie oprogramowania odpowiadającego funkcjonalnie oprogramowaniu firmy DEC, tj. wzbogacenie systemu operacyjnego o bibliotekę IBLIB (opisaną w p. 2.3) i handler interfejsu IEC, umożliwiające pisanie programów w FORTRAN-ie.

Ostateczna ocena przydatności poszczególnych wariantów nie jest łatwa. Na ocenę taką niewątpliwie mają wpływ czynniki związane z konkretnym systemem pomiarowym oraz indywidualne cechy jego użytkownika, np.:

- konfiguracja systemu pomiarowego,
- wymagania dotyczące prędkości transmisji,
- parametry techniczne zastosowanych przyrządów pomiarowych,

- wymagania dotyczące sposobu przetwarzania danych w komputerze,
- wiedza fachowa i doświadczenie użytkownika,
- posiadane zasoby programowe,
- wielkość środków finansowych na rozwój systemu.

Na zakończenie warto wspomnieć o układach VLSI znajdujących za granicą coraz powszechniejsze zastosowanie jako kontrolery magistrali IEC. Układy te maksymalnie upraszczają połączenie jednostki centralnej (np. mikrokomputera) z innymi urządzeniami stanowiącymi system pomiarowy. Zastosowanie elementów VLSI pozwala ponadto osiągnąć duże prędkości transmisji. Opis wspomnianych układów można znaleźć w [6, 8].

LITERATURA

- [1] IEEE Standard 488-1975, Digital interface for programmable instrumentation - The IEEE, Inc. N.Y. 1975.
- [2] Łączyński J.: Szeregowo-bajtowy interfejs dla urządzeń pomiarowych IEC-625. Biuletyn Techniczny Mera. Lipiec 1979.
- [3] Digital instrument course. Part 4 - IEC-BUS Interface. N.V. Philips' Gloeilampenfabrieken Test and Measuring Department Eindhoven, The Netherlands.
- [4] Hewlett-Packard Interface Bus User's Guide for systems based on the HP9830A Calculator. November 1974.
- [5] Oprogramowanie interfejsu IEC-625, ISS, Zakład programowania, Katowice, czerwiec 1982.
- [6] Korbziel E.: Problemy i realizacja interfejsu IEC w układach mikroprocesorowych, Zeszyty Naukowe Politechniki Śląskiej, Gliwice 1981.
- [7] Forbes B.E.: IEEE 488: A proposed microcomputer I/O bus standard. Computer Design, November 1978.
- [8] Williams R.M.: LSI Chips ease standard 488 Bus interfacing. Computer Design, October 1979.
- [9] Ghannam A.N., Fayed M.M.: Unified Approach to designing hardware based on IEEE STD 488. Computer Design, September 1979.
- [10] Opracowanie dokumentacji systemu operacyjnego mikrokomputera MERA-80. Zakład Systemów Mikrokomputerowych ISS, Sosnowiec 1980.
- [11] LSI-11 (PDP-11)03 processor handbook. DEC 1975.
- [12] Microcomputer handbook series: "Memories and peripherals". 1978/79.

Recenzenci: Mgr Danuta Tábicka, Doc. dr inż. Jerzy Łączyński

Wpłynęło do Redakcji 17.12.1982 r.

ПРОГРАММНЫЕ ВОЗМОЖНОСТИ ПРИСПОСОБЛЕНИЯ ЭВМ ТИПА MERA-60
ДЛЯ УПРАВЛЕНИЯ ИНТЕРФЕЙСОМ IEC-625

Р е з ю м е

В статье представлено проблемы программного контроля сообщения с применением интерфейса IEC-625 (IEEE-488) в измерительных системах с ЭВМ исполняющей роль единственного контролера. В качестве примера представлено три решения зарубежных фирм. Предложено решение для ЭВМ типа MERA-60. Проведено оценку описанных решений с точки зрения пригодности для потребителя.

SOFTWARE ADAPTABILITY OF THE MERA-60 COMPUTER TO CONTROL
THE IEC-625 INTERFACE

S u m m a r y

Problems of software control of communication through the IEC-625 (IEEE-488) interface in measuring systems with a computer as a controller are dealt with in the paper.

Three foreign firms exemplary realisations are introduced. Some suggestions of the solution for the MERA-60 are presented next. Finally the solutions described in the paper are evaluated from the point of view of their usefulness.

