

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW

440

PIAP

A

Al. Jerozolimskie 202

02-486 Warszawa

Zespół Zrobotyzowanych Systemów Inteligentnych ZSI

Główny wykonawca: mgr inż. Zbigniew Pilat

Wykonawcy: doc. dr inż. Piotr Jabłoński
mgr inż. Marek Pachuta
mgr inż. Zbigniew Wawerek

Nr zlecenia: S1667

Tytuł pracy: Aspekty sterowania robotami ciężkimi na przykładzie URP-120

Numer i tytuł etapu: Etap 3. Opracowanie wyników i wniosków z badań. Weryfikacja dokumentacji i oprogramowania robota URP-120.

Zlecaniodawca: praca statutowa PIAP

Pracę rozpoczęto dnia: 1996.11.01.

zakończono dnia: 1996.12.31.

Kierownik Ośrodka


mgr inż. Zbigniew Pilat

Dyrektor Pionu


dr inż. Jan Jabłkowski

Praca zawiera:

stron 6
rysunków --
fotografii --
tabel --
tablic --
załączników 9

Rozdzielnik - ilość egz.:

Egz. 1 ZSI
Egz. 2 ZSS
Egz. 3 OIN
Egz. 4
Egz. 5
Egz. 6

Nr rej. 7402

Analiza deskryptorowa:

ROBOTY PRZEMYSŁOWE, STEROWANIE, OPROGRAMOWANIE, BADANIA

Analiza dokumentacyjna:

Sprawozdanie zawiera opis prac zrealizowanych w ramach weryfikacji po badaniach dokumentacji i oprogramowywania robota URP-120.

Tytuły poprzednich sprawozdań:

Aspekty sterowania robotami ciężkimi na przykładzie URP-120. Etap 2. Uruchomienie, oprogramowanie i badania robota URP-120. spraw. PIAP nr rej. 7401.

1. Wstęp

W ramach dotychczasowych prac nad robotem URP-120 wykonano następujące zadania:

- a) Montaż, zestawienie i uruchomienie w zakresie sprzętu układu sterowania URP w wersji dostosowanej do robota o udźwigu 120 kg.
- b) Testy części napędowej i obiektowej układu sterowania URP-120.
- c) Uruchomienie kanału transmisji szeregowej RS232C w jednostce centralnej układu sterowania URP-120.
- d) Posadowienie prototypowej wersji podstawowego programu sterującego robota URP-120. Testowanie funkcjonalne układu sterowania robota URP-120.
- e) Połączenie układu sterowania z częścią manipulacyjną i uruchomienie kompletnego robota URP-120.
- f) Wykonanie i uruchomienie programu sterującego dla robota URP-120.
- g) Przeprowadzenie podstawowych prób funkcjonalnych.

W efekcie tych prac powstał prototyp robota URP-120. Po badaniach zweryfikowano dokumentację w zakresie sprzętowej układu sterowania i w zakresie oprogramowania. Przygotowano wzorce pamięci EPROM. Zebrano też szereg uwag i wniosków, które będą przydatne generalnie przy opracowaniu koncepcji i realizacji dalszych prac nad robotami URP. Niniejsze sprawozdanie przedstawia właśnie wyniki obserwacji i płynące z nich wnioski, jest podsumowaniem całej pracy nad robotem URP-120.

2. Wyniki i wnioski z prac uruchomieniowych i badań robota URP-120.

Wiele uwag i wniosków nasunęło się już w trakcie prac uruchomieniowych układu sterowania i całego robota URP-120. Przeprowadzone próby dały dodatkowe obserwacje. Dla uporządkowania zostały one zebrane w poniższych punktach.

I. Niezawodność układu sterowania URP. Prace nad układem sterowania URP-120 i później przy uruchamianiu całego robota pokazały, że sterownik URP cechuje bardzo wysoka niezawodność. Układ, na którym prowadzono prace był adaptowany z prototypu układu sterowania URP-60. Wcześniej, właśnie z częścią manipulacyjną 60 kg, przeszedł cały cykl badań robota. Można więc powiedzieć, że był to zestaw w pewnym stopniu wyeksploatowany. Pomimo tego, podczas prac nad URP-120, nie było praktycznie żadnych kłopotów sprzętowych, awarii, niesprawności itp. *Wniosek - należy trzymać się przyjętej koncepcji konstrukcji sterowania URP i konsekwentnie ją rozwijać, nie szukając całkiem nowych rozwiązań.*

II. Okablowanie części manipulacyjnej URP-120. Podstawowym powodem przedłużenia prac nad uruchomieniem całego robota były problemy z częścią manipulacyjną. Kilkakrotnie doszło do uszkodzenia okablowania. Jest ono przy tym, w wielu miejscach poprowadzone dość nieszczęśliwie. Przykładowo wysoce awaryjne jest poprowadzenie przewodów wokół osi obrotu piątego stopnia swobody. Są one mocowane na sprężynie, która odwija się i zwija wraz z ruchem tej osi. Sprężyna ta ma dość trudne warunki pracy i często ulega uszkodzeniu. Poza tym, aby to rozwiązanie prawidłowo działało, przewody muszą się równo układać, co po pewnym okresie eksploatacji jest praktycznie niemożliwe. *Wniosek - jedną z podstawowych prac przy modernizacji części manipulacyjnej robota URP-120 powinna być zmiana prowadzenia okablowania, np. stosując rozwiązania opracowane i sprawdzone w robocie 120 kg z układem sterowania Bosch.*

III. Moc napędów robota. Podczas badań robot wykonywał instrukcje ruchu z zadeklarowanymi maksymalnymi prędkościami. Widać jednak było, że prędkości tych nie osiągał. Po prostu było to niemożliwe przy aktualnym możliwym do uzyskania przyspieszeniu, na konstrukcyjnie ograniczonych zakresach. Dodatkowo, przy stosowanych napędach robot jest czuły na wielkość ciśnienia w cylindrach odciążenia pneumatycznego. *Wniosek - dla realnych zastosowań przemysłowych należy opracować zastosowanie mocniejszych silników.*

- IV. Podtrzymanie bateryjne.** Podczas prac miało miejsce wyczerpanie się baterii podtrzymujących program użytkowy na jednostce centralnej. Nastąpiło to podczas postoju robota, gdy był wyłączony. Baterie rzeczywiście były stare i miały prawo się wyczerpać lecz operator nie mógł przewidzieć momentu zaistnienia tego faktu. *Wniosek - trzeba wprowadzić sygnalizację stanu naładowania baterii.*
- V. Trwałość części mechanicznej.** W ramach badań przeprowadzono próbę pracy ciągłej. Trwała ona tylko 8 godzin, co dla celów realizowanego zlecenia wystarczyło. Nie dało jednak żadnych informacji na temat trwałości części mechanicznej. Biorąc pod uwagę ogólnie wysoką niezawodność układu sterowania, URP potwierdzoną w innych zastosowaniach, wydaje się, że krytyczną dla trwałości całego robota URP-120 jest niezawodność części manipulacyjnej. *Wniosek - w razie kontynuowania prac nad robotem URP-120 należy przeprowadzić długotrwałą (rzędu 800-1000 godzin) próbę zmęczeniową, celem znalezienia słabych punktów mechaniki robota.*

3. Możliwości zastosowania robota URP-120.

Biorąc pod uwagę parametry techniczne robota URP-120, jego cechy, które ujawniły się podczas przeprowadzonych prac uruchomieniowych i badań oraz aktualne rozeznanie na rynku można wskazać aktualne, możliwe, realne do praktycznych przedsięwzięć, obszary zastosowań robota URP-120.

- I. **Paletyzowanie.** Do tego zastosowania robot wymagałby najmniejszych przeróbek. Oczywiście na początek należałoby szukać aplikacji, gdzie nie jest wymagany krótki czas cyklu, a więc i duże przyspieszenia/opóźnienia napędów robota. Dlatego pod kątem tego obszaru trzeba się skupić na zastosowaniach w środowisku uciążliwym dla człowieka lub tam, gdzie jest wymagany wyjątkowo duży udźwig, przekraczający możliwości człowieka. Do takich obszarów można zaliczyć paletyzowanie proszków do prania (duże pudełka są pakowane po kilkanaście sztuk - najczęściej w folię), przenoszenie transporterów z napojami (palety mają po kilka warstw, o całkowitej wysokości rzędu 180-200 cm), przenoszenie artykułów workowanych (mogą być to worki o całkowitej masie rzędu 50 do 100 kg - przemysł spożywczy, nawozy).
- II. **Czyszczenie ciężkich detali.** Czyszczenie odlewów od lat należy do popularnych aplikacji robotów przemysłowych na świecie. W naszym kraju proces ten nie jest chętnie robotyzowany z uwagi na wciąż niskie koszty pracy ręcznej w stosunku do kosztów instalacji zautomatyzowanych. Są jednak wyroby o dużych gabarytach i dużym ciężarze, którymi człowiekowi po prostu trudno jest manipulować. Tu istnieje szansa wejścia z robotem URP-120. Przykładem może być asortyment produkcji niektórych odlewni (korpusy maszyn) oraz wyroby typu felgi samochodowe.
- III. **Zgrzewanie punktowe.** Z uwagi na swój udźwig robota URP-120 może śmiało operować zgrzewarką z transformatorem umieszczonym na ramieniu, bezpośrednio przy kleszczach. Ponieważ robotyzacja tej operacji jest opłacalna prawie wyłącznie przy produkcji długich serii wyrobów, wydaje się, że możliwości wdrożeń w tym obszarze należy szukać przede wszystkim w motoryzacji.

Podejmując działania akwizycyjno-promocyjne w ww obszarach trzeba pamiętać, że posiadany, wykorzystywany w próbach prototyp robota URP-120 nie nadaje się do zastosowania w warunkach przemysłowych, do pracy w reżimie produkcyjnym. W razie podjęcia tematu aplikacyjnego musi być wykonany nowy egzemplarz. Dodatkowo takie pierwsze wdrożenie będzie zarazem poletkiem doświadczalnym, na którym wyjdą wszelkie niedostatki konstrukcji., Biorąc to pod uwagę oraz cenę samego robota wydaje się, że realne jest jedynie przeprowadzenie pierwszego

wdrożenia przy wsparciu dofinansowania zewnętrznego, np. w ramach projektu celowego KBN.

Załącznik 1 - opis dyskietki z oprogramowaniem robota URP-120.**URP-120**

Dyskietka zawiera kompletny program sterujący dla robota URP-120 według stanu na dzień 23 grudnia 1996. Program został zarchiwizowany przy wykorzystaniu narzędzi Norton Commander ver. 4.0 - metoda kompresji 'ARJ'. Do rozpakowania należy użyć narzędzi dekompresji 'ARJ' pakietu Norton Commander, z opcją odtworzenia katalogów. Po rozpakowaniu otrzymujemy katalog '_120', wraz z podkatalogami. Zapisane są w nich:

- pliki źródłowe,
- biblioteki,
- program w postaci wynikowej, ładowalnej do EPROM - 'ROMFILE.HEX',
- program inicjalizacji pakietu jednostki centralnej MV-52, w postaci wynikowej, ładowalnej do EPROM, w dwóch wersjach:
 - z monitorem - 'MONITON.HEX',
 - bez monitora - 'MONITOFF.HEX',

Na dyskietce znajduje się również zbiór URPEPROM.DOC - zapisane w edytorze Word for Windows 2.0 zasady tworzenia wzorców pamięci EPROM z programem sterującym robotów URP dla jednostki centralnej MV-52, przy wykorzystaniu programatora SEPROG.

Załącznik 2 - Zasady tworzenia wzorców pamięci EPROM z programem sterującym robotów URP, przy wykorzystaniu programatora SEPROG.

1. Zrobić plik ROMFILE.HEX
2. Ze zbioru ROMFILE.HEX usunąć, np. przy pomocy edytora, następujące rekordy:

START_ADDRESS_RECORD - typ rekordu '03'

END_OF_FILE_RECORD - typ rekordu '01'

(są to dwa ostatnie rekordy pliku ROMFILE.HEX).

3. Skopiować plik z kodem programu sterującego (ROMFILE.HEX) i plik z kodem programu 'monitor POS' (MONITON.HEX jeśli chcemy aby monitor zgłaszał się w zależności od ustawienia zwory na płycie złącza układu sterowania lub MONITOFF.HEX jeśli monitor nie jest nam potrzebny) do jednego pliku, z którego będziemy robili wzorce pamięci EPROM:

COPY ROMFILE.HEX + MONITOR.HEX ROM.HEX

4. Przejść do katalogu SEPROG i skopiować do niego plik ROM.HEX.
5. Uruchomić program SEPROG.EXE
6. Wybrać polecenie LOAD z menu FILE.
7. Jako zbiór do ładowania podać ROM.HEX - pokaże się okienko z pytaniami:
 - podać File offset - E0000,
 - wybrać Loaded bytes - Even!!!,
 - najechać myszą na OK i kliknąć.

8. Wybrać polecenie PROGRAM z menu DEVICE, sprawdzić, że:

Device Start Address = 00000

Device End Address = 0FFFF

wstawić do podstawki programatora czystą kość EPROM 27C512, opisaną jako "Low", najechać myszą na OK i kliknąć. Programowanie trwa ok. 7-8 minut.

9. Wybrać polecenie LOAD z menu FILE.
10. Jako zbiór do ładowania podać ROM.HEX - pokaże się okienko z pytaniami:
 - podać File offset - E0000,
 - wybrać Loaded bytes - Odd!!!,
 - najechać myszą na OK i kliknąć.

11. Wybrać polecenie PROGRAM z menu DEVICE, sprawdzić, że:

Device Start Address = 00000

Device End Address = 0FFFF

wstawić do podstawki programatora czystą kość EPROM 27C512, opisaną jako "High", najechać myszą na OK i kliknąć.

12. Wybrać polecenie QUIT z menu FILE. Wyłączyć programator.

Załącznik 3 - *Specyfikacja dokumentacji konstrukcyjnej układu sterowania robota URP-120*Dokumentacja konstrukcyjna:

Numer	Tytuł dokumentacji
8119	Cyfrowy sterownik położenia osi MV20
8120	Zasilacz rezolwerów MV21
8123	Adapter panelu programowania
8124	Panel operacyjny
8125	Zespół bezpieczników i styczników
8126	Kabel robota
8127	Przylącze wejść/wyjść
8128	Panel programowania
8130	Płytki sterowania panelu programowania
8131	Kabel panelu programowania
8134	Zespół zasilaczy
8183	Prostownik układu zasilania
8186	Zespół hamulca URP-120

Dokumenty związane:

- Dokumentacja techniczno-ruchowa układu sterowania robotów URP-60
- Program badań układu sterowania robotów URP-60
- Dokumentacja techniczno-ruchowa panelu programowania
- Warunki technicznego odbioru panelu programowania

Załącznik 4 - *Specyfikacja oprogramowania układu sterowania robota URP-120***Katalog główny '_120'**

- RESTART.A86
- COP.BAT
- COP_ALL.BAT
- GO.BAT
- LIB.BAT
- LIBRARY.BAT
- AXESCNTR.C86
- ERROR.C86
- ERRTXT.C86
- GLOBVAR.C86
- GRIPOPER.C86
- HARDERR.C86
- INITMAIN.C86
- MAIN.C86
- SETSYSST.C86
- SWTCHEXT.C86
- SYSTEM.C86
- TIMELAG.C86
- VERSION.C86
- ERRORS.H
- EXTWORLD.H
- MAIN.H
- SETJMP.H
- MAIN.LIB

Podkatalog EPROM

- ROMFILE.BAT
- ROMLOC.CON
- PROFILE.KED
- ROMFILE.MP1
- ROMFILE.HEX
- ROMFILE.MP2

Podkatalog EXTMOV

- LIB.BAT
- LIBRARY.BAT
- INV_TOR.C86
- MOVE.C86
- CALCEXT.C86
- DEFTOOL.C86
- DTOI.C86
- DTOL.C86
- EXTCONST.C86
- INITEXTM.C86

- INTORAD.C86
- LINEAR.C86
- M_TOR_2.C86
- MAKE_PP.C86
- PARLIN.C86
- RADTOIN.C86
- STINTERP.C86
- EXTMOV.H
- INDEKS.H
- LINEAR.H
- MOVE.H
- EXTMOV.LIB

Podkatalog GLOBSPAC

- LIB.BAT
- LIBRARY.BAT
- CLRPRMLP.C86
- DELINS.C86
- DELTOOL.C86
- FINDINS.C86
- FINDTOOL.C86
- GETCURR.C86
- GETEXT.C86
- GETFRST.C86
- GETINTR.C86
- GETLAST.C86
- GETMIDD.C86
- GETNEXT.C86
- GETPREV.C86
- GLOBSINI.C86
- GLOBSVAR.C86
- INIT_GS.C86
- PUTINSNO.C86
- SEARCH.C86
- SEARCHIN.C86
- SEARPREV.C86
- SETPRMLP.C86
- SHIFT.C86
- STOREINS.C86
- STORTOOL.C86
- UPDATIDX.C86
- VELCOEFF.C86
- POPPOS.C86
- CONST.H
- GLOBSPAC.H
- GLOB_RAM.LIB

—	GLOBSIZE.A86	—	UNEX_INT.A86
—	GLOB_ROM.LIB	Podkatalog INTMOV	
—	GLOBSPAC.LIB	—	LIB.BAT
Podkatalog 'HARDWARE'		—	LIBRARY.BAT
—	LIB.BAT	—	QUASILIN.C86
—	LIBRARY.BAT	—	DIVIDE.A86
—	CSRA.A86	—	INITINTM.C86
—	DIS_IN.A86	—	INRANGE.C86
—	ENABL_IN.A86	—	INTMCONS.C86
—	HARD_CON.A86	—	INTMVAR.C86
—	INBYTE.A86	—	MICROSTP.C86
—	INI_186.A86	—	SENDABS.C86
—	INI_8259.A86	—	SIMPLSTP.C86
—	INI_Z853.A86	—	SIVCNSTR.C86
—	INT_TIME.A86	—	ARM.H
—	INT_Z853.A86	—	INTMOV.H
—	INWORD.A86	—	MICROSTP.H
—	OPOZNIEN.A86	—	QUASILIN.H
—	OUTBYTE.A86	—	SENDABS.H
—	OUTWORD.A86	—	INTMOV.LIB
—	PARAMTR.A86	Podkatalog INTRPRTR	
—	SET_VECT.A86	—	LIB.BAT
—	UNEX_INT.A86	—	LIBRARY.BAT
—	AXISSTOP.C86	—	INTRRPTD.C86
—	BLINK.C86	—	CLEAN.C86
—	COUNTER2.C86	—	DOEXTRA.C86
—	CTRLRDY.C86	—	DOINSTR.C86
—	EXTINGSH.C86	—	INIINTRP.C86
—	GETCLOCK.C86	—	INTRPCNS.C86
—	GRIPPERS.C86	—	INTRPRTR.C86
—	HARDCONS.C86	—	INTRPVAR.C86
—	HARDVAR.C86	—	IS.C86
—	INITHARD.C86	—	JUMPCOND.C86
—	INPOS.C86	—	VALUE1B.C86
—	LIGHT.C86	—	WAITCOND.C86
—	READAPOS.C86	—	WAITSTAT.C86
—	SENDINC.C86	—	WTFINPOS.C86
—	SNDCORIN.C86	—	INSCODES.H
—	STOPHSYN.C86	—	INSTRUCT.H
—	STRTHSYN.C86	—	INTRPRTR.H
—	SYNCHRND.C86	—	TYPES.H
—	USERIN.C86	—	WAITCOND.H
—	USEROUT.C86	—	INTRPRTR.LIB
—	HARDWARE.H	Podkatalog INSTR	
—	HARDWARE.LIB	—	LIB.BAT
Podkatalog NORM_INT		—	LIBRARY.BAT
—	UNEX_INT.A86	—	CALL.C86
Podkatalog TEST_INT		—	DEFV.C86

-	EMPTY.C86	-	ROTMATRIX.C86
-	ENDPGM.C86	-	VECTANGL.C86
-	GRIP.C86	-	VECTMDSQ.C86
-	JUMP.C86	-	VECTNRML.C86
-	JUMPEQ.C86	-	VECTSUB.C86
-	JUMPNE.C86	-	MATVMUL.C86
-	LOOP.C86	-	SCLMUL.C86
-	LPEND.C86	-	VECTADD.C86
-	NOOP.C86	-	VECTMUL.C86
-	POSC.C86	-	MATH87.H
-	POSL.C86	-	MATH87.LIB
-	POSQ.C86	Podkatalog OPERPNL	
-	RET.C86	-	LIB.BAT
-	SET.C86	-	LIBRARY.BAT
-	STRPGM.C86	-	OPER_TAB.A86
-	SUBR.C86	-	AUTOWORK.C86
-	TOOL.C86	-	DONOTHNG.C86
-	WAIT.C86	-	EMERSTOP.C86
-	WAITEQ.C86	-	GO_ON.C86
-	WAITNE.C86	-	LOADPGM.C86
-	INSTR.H	-	MANUWORK.C86
-	POSCORI.C86	-	OPERPNL.C86
-	INSTR.LIB	-	PGMSTART.C86
Podkatalog MANMOV		-	STOPERR.C86
-	LIB.BAT	-	STOPPGM.C86
-	LIBRARY.BAT	-	STOPSYN.C86
-	CARFRAME.C86	-	SYNCHRNZ.C86
-	INITMANM.C86	-	OPERPNL.LIB
-	MANMOV.C86	Podkatalog PROGPNL	
-	INTFRAME.C86	-	LIB.BAT
-	MANMCONS.C86	-	LIBRARY.BAT
-	MANMVAR.C86	-	DECINSR.A86
-	PGMBUTT.C86	-	ATOD.C86
-	SWITCHFR.C86	-	CHKEYS.C86
-	MANMOV.H	-	CLRLINE.C86
-	MANMOV.LIB	-	D_EMPTY.C86
Podkatalog MATH87		-	D_STRANG.C86
-	LIB.BAT	-	DECINSTR.C86
-	LIBRARY.BAT	-	DECNPAR.C86
-	FARCCOS.A86	-	DIODEOFF.C86
-	FATAN2.A86	-	DIODEON.C86
-	FCOSSIN.A86	-	DTOA.C86
-	FRNDINT.A86	-	ESDOFF.C86
-	FSQRT.A86	-	NUMBER.C86
-	MATHCONS.A86	-	OPTCHNG.C86
-	VECTCOPY.A86	-	OPTREAD.C86
-	VECTMOD.C86	-	PANELINI.C86
-	CIRCPAR.C86	-	PANELVAR.C86

-	READNUM.C86	-	D_ENDPGM.C86
-	RESTCURS.C86	-	D_ENDREP.C86
-	SAVECURS.C86	-	D_GRIPP.C86
-	SETCURS.C86	-	D_JUMP.C86
-	TCBTAB.C86	-	D_OUFLAG.C86
-	WAITFKEY.C86	-	D_REPEAT.C86
-	BUTTONS.H	-	D_RET.C86
-	CHKEY.H	-	D_STRPGM.C86
-	CURSOR.H	-	D_SUBR.C86
-	DECINSTR.H	-	D_TOOL.C86
-	DIODES.H	-	D_VELOC.C86
-	NUMBER.H	-	D_WAIT.C86
-	PROGPNL.H	-	DECOBO.C86
-	PROGPNL.LIB	-	DECREL.C86
Podkatalog EDIT		-	OTHERINS.C86
-	LIB.BAT	-	P_CALL.C86
-	LIBRARY.BAT	-	P_ENDPGM.C86
-	EDIT_TAB.A86	-	P_ENDREP.C86
-	BACKWARD.C86	-	P_GRIPP.C86
-	DELETE.C86	-	P_JUMP.C86
-	EDIT.C86	-	P_OUFLAG.C86
-	FORWARD.C86	-	P_REPEAT.C86
-	PAREDIT.C86	-	P_RET.C86
-	RENUMER.C86	-	P_STRPGM.C86
-	SETINSNO.C86	-	P_SUBR.C86
-	DELETE.H	-	P_TOOL.C86
-	POSEDIT.C86	-	P_VELOC.C86
-	EDIT.LIB	-	P_WAIT.C86
Podkatalog MANOPER		-	PGMOBO.C86
-	LIB.BAT	-	PGMREL.C86
-	LIBRARY.BAT	-	OTHERINS.H
-	MANO_TAB.A86	-	OTHERINS.LIB
-	ACTPOS.C86	Podkatalog POSINS	
-	FREEMEMO.C86	-	LIB.BAT
-	KOMPUTER.C86	-	LIBRARY.BAT
-	MANOPER.C86	-	D_POS.C86
-	MEMSERV.C86	-	P_POS.C86
-	SETMVELO.C86	-	POSINS.C86
-	SETREFPO.C86	-	PUSHPOS.C86
-	STOREREF.C86	-	POS.H
-	TOOLDEF.C86	-	POSINS.LIB
-	USER_IO.C86	Podkatalog START	
-	MANOPER.LIB	-	LIB.BAT
Podkatalog OTHERINS		-	LIBRARY.BAT
-	LIB.BAT	-	CLEARPGM.C86
-	LIBRARY.BAT	-	SETAVEL.C86
-	OTHE_TAB.A86	-	SETSYMUL.C86
-	D_CALL.C86	-	START.C86

-	STARTPER.C86	-	READEEPR.A86
-	STARTPGM.C86	-	READ_MAS.A86
-	STEPPGM.C86	-	RENAME.A86
-	START.LIB	-	SEND.A86
Podkatalog STDIO		-	SRCHNUMB.A86
-	LIB.BAT	-	STALE.A86
-	LIBRARY.BAT	-	TABLICE.A86
-	IOINIT.C86	-	WRITEEPR.A86
-	IOVARIAB.C86	-	WRITE_MA.A86
-	READ.C86	-	READMEM.C86
-	STREAMS.C86	-	WRITEMEM.C86
-	UNLOCK.C86	-	MASMEM.LIB
-	WRITE.C86	Podkatalog TEST	
-	BUFFERS.H	-	FORMAT.A86
-	CHARCODE.H	-	WRITEEPR.A86
-	SERIAL.H	Podkatalog PANEL	
-	STDIO.H	-	LIB.BAT
-	STDIO.LIB	-	LIBRARY.BAT
Podkatalog KONSOLA		-	BLANKPNL.A86
-	LIB.BAT	-	BYTEPNL.A86
-	LIBRARY.BAT	-	CLRLINE.A86
-	BLANK.A86	-	CURSLEFT.A86
-	BYTE_CON.A86	-	DECBYTPN.A86
-	CONSTANS.A86	-	DECDPNL.A86
-	DECDOUBL.A86	-	DECWORPN.A86
-	DEC_BYTE.A86	-	GC_PNL_A86
-	DEC_WORD.A86	-	GC_WAIT_A86
-	GETCH_EX.A86	-	GETCHPNL.A86
-	GETCH_WA.A86	-	MESSPNL.A86
-	MESSAGE.A86	-	PRCHAR_A86
-	NEW_LINE.A86	-	PRSTR_A86
-	PUTCH_EX.A86	-	PUTCHPNL.A86
-	WORD_CON.A86	-	REJESPNL.A86
-	KONSOLA.LIB	-	WORDPNL.A86
Podkatalog MASMEM		-	ESC_SERV.C86
-	LIB.BAT	-	GC_PANEL.C86
-	LIBRARY.BAT	-	MKIMAGE.C86
-	BRAK_PLK.A86	-	MOVEKPAD.C86
-	CALCADDR.A86	-	PC_PANEL.C86
-	DEL_PROG.A86	-	POPIMAGE.C86
-	DIR.A86	-	PR_CHAR.C86
-	FORMAT.A86	-	PR_STR.C86
-	INTRKLAW.A86	-	PSHIMAGE.C86
-	LOAD.A86	-	PUTTOBUF.C86
-	NAZWA.A86	-	RD_PANEL.C86
-	PAR_TRAN.A86	-	WR_PANEL.C86
-	POJEMN.A86	-	IMAGE.H
-	PROTECT.A86	-	PANEL.LIB

Podkatalog RETARGER

- LIB.BAT
- LIBRARY.BAT
- LSEEK.A86
- MALLOC.A86
- MAP_LEN.A86
- SEMAPHOR.A86
- CONIO.C86
- EXIT.C86
- ISATTY.C86
- STDIO.C86
- STDOPEN.C86
- THREAD.C86
- RETARGER.LIB

Podkatalog TO_RAM

- RAMFILE.BAT
- RAMLINK.BAT
- RAMLOC.BAT
- RAMFIL.BAT
- RAMLINK.CON
- RAMLOC.CON
- RAMFILE.MP1
- RAMFILE.HEX
- RAMFILE.MP2

Podkatalog 'UTILITY'

- LIB.BAT
- LIBRARY.BAT
- ADDMEM.A86
- ADDRESS.A86
- CHAR_INT.A86
- DUMP.A86
- INCRMEM.A86
- INT_CHAR.A86
- LAMPKI.A86
- MONITOR.A86
- MOV_BYTE.A86
- PARAMETR.A86
- PAUSE.A86
- POINTEXT.A86
- REJESTRY.A86
- UTILITY.LIB