

# Programska implementacija modula za obradu ulazno/izlaznih signala kod robotskog kontrolera opšte namene

Vladan DEVEDZIC  
Branko KARAN  
Milija TIMOTIJEVIĆ  
Institut "Mihajlo Pupin"  
Beograd

Primljeno: 27.06.1987.g.

## SAŽETAK

U radu je opisan programski modul za obradu ulazno/izlaznih signala kod industrijskog robotskog kontrolera opšte namene. Data je koncepcija asinhronog praćenja promenljivih spoljnih signala i uslova uz istovremeno izvršavanje sistema sa okolinom. Razmotrene su različite mogućnosti pristupa ulazno/izlaznim signalima iz korisničkih programa napisanih na robotskom programskom jeziku RL, koji je ugrađen u kontroler. Takođe su opisane i mogućnosti rekonfiguracije ulazno/izlaznog modula sistema u slučaju promene radnog okruženja ili vrste aplikacije. U diskusiji o trenutnim performansama realizovanog sistema sa stanovišta obrade ulazno/izlaznih signala, ukazano je na mogućnosti nadgradnje i proširenja sistema.

## ABSTRACT

The program module for input/output (I/O) signal processing in the general purpose industrial robot controller is described. The concept of asynchronous scanning of external signals and conditions running simultaneously with the robot program is used as a method of robot system communication with the environment. Various alternatives of accessing I/O signals from user programs, written in the controller's built-in robot programming language are also considered. The I/O module reconfiguration capabilities are described for the case of various application and/or environment changes. The system extension possibilities are discussed from the I/O signal processing viewpoint.

## 1. UVOD

Robotski kontroler opšte namene, realizovan u

Institutu "Mihajlo Pupin", predstavljen je u više ranijih radova (/1/, /2/, /3/, /4/, /5/). Ovaj kontroler se može primeniti za upravljanje ne-redundantnim manipulatorom proizvoljnog tipa, pokretnim DC elektro motorima ili hidrauličkim aktuatorima. Kontroler je realizovan u vidu multiprocesorskog sistema sa tri 16-bitna procesora INTEL 8086. Hardverska podrška sistema uključuje i module za A/D i D/A konverziju, paralelne digitalne ulazno/izlazne (U/I) module, serijske U/I module, itd. Periferijski uređaji kontrolera obuhvataju video terminal, štampač, upravljačku palicu i dr.

Softver koji je ugrađen u sistem sastoji se od više međusobno povezanih modula: modul za sračunavanje kinematike robota u realnom vremenu, modul za kontrolu digitalnih servosistema i izračunavanje dinamike u realnom vremenu, modul za programiranje robota, interpretator robotskih programa, modul za obradu U/I signala, modul za komunikaciju sa operatorom, modul za specifikaciju parametara robota i drugih delova sistema i modul za upravljanje robotom korišćenjem upravljačke palice.

Robotski programski jezik, RL, ugrađen u sis-

tem, omogućuje programiranje robota putem kreiranja i izvršavanja robotskih programa. To je proceduralni jezik visokog nivoa. Robotski jezici tog tipa podrobno su opisani u /7/, /8/, /9/. Robotski program predstavlja niz komandi robotskog jezika kojima se opisuje neki zadatak koji robot izvršava. Robotski programi napisani u programskom jeziku RL mogu da sadrže razne vrste podataka (celobrojne, realne, logičke, itd.), kao i razne vrste aritmetičkih i logičkih izraza. Posebna grupa komandi namenjena je za zadavanje kretanja robota, kako u spoljašnjim, tako i u unutrašnjim koordinatama (komande tipa MOVE). Programski jezik RL takodje omogućuje i kreiranje potprograma, grananje u programima, obradu spoljašnjih signala i uslova, prekidanje izvršavanja programa, štampanje podataka, itd.

Modul za obradu U/I signala (u daljem tekstu U/I modul), predstavlja posebnu celinu u ovom programskom sistemu. Zbog svoje specifične uloge, ovaj modul predstavlja jedan od kritičnih modula sistema u pogledu brzine rada. Modul je relativno nezavisan od ostalih modula sistema, nužno je povezan sa hardverskim i perifernim uređajima, a njegovim radom se upravlja preko pojedinih komandi robotskog jezika.

Zadatak U/I modula robotskog kontrolera opšte namene je ostvarivanje komunikacije sa okolinom. Preko ovog modula robotski kontroler obezbeđuje ne samo učitavanje i generisanje sinhronizacionih signala i razmenu informacija, već i praćenje promenljivih spoljnih podataka, signala i uslova paralelno sa izvršavanjem osnovnog manipulacionog zadatka /3/. To znači da U/I modul ima dva osnovna načina rada: sinhroni i asinhroni.

Sinhrono učitavanje signala iz okoline i generisanje signala koji se šalju prema perifernim uređajima vrši se onda kada se to neposredno zahteva preko određene komande u robotskom programu. Za vreme izvršavanja takve komande sistem ne obavlja nikakav drugi zadatak.

S druge strane, korisnik u programu može da zahteva da se robotski zadatak i dalje odvija, ali da se istovremeno kontroliše da li je nastupio neki događaj u sistemu ili u radnom okruženju koji iziskuje posebnu pažnju, odnosno posebnu obradu. U tom slučaju U/I sistem počinje sa praćenjem određenih spoljnih uslova,

putem ispitivanja odgovarajućih U/I signala, u cilju detektovanja da li je takav događaj nastupio. Ovo praćenje spoljnih uslova U/I modul obavlja potpuno asinhrono, nezavisno od izvršavanja robotskog programa. Asinhrono ispitivanje spoljašnjih uslova obavlja se tako što u regularnim vremenskim trenucima program U/I modula privremeno prekida rad interpretatora robotskog programa i skanira nekoliko uslova za liste zadatah, posle čega se nastavlja interpretacija robotskog programa od mesta prekida. U slučaju da se ispuni neki od praćenih uslova, U/I modul šalje interpretatoru odgovarajuću informaciju /3/, /4/, /5/, /6/.

## 2. KONCEPCIJA OBRADU U/I SIGNALA

Robotski kontroler je namenjen za upravljanje radom robota koji se može nalaziti u raznovrsnim tehnološkim sredinama. Svaka tehnološka sredina predstavlja specifično radno okruženje čije stanje u svakom trenutku karakterišu vrednosti raznih fizičkih parametara (temperatura, pritisak, dimenzije i masa pojedinih predmeta, itd.), trenutni raspored pojedinih objekata u radnom prostoru, trenutni položaj manipulatora u odnosu na druge objekte i na radni predmet, razne sile i momenti koji deluju na robot i radni predmet, itd. Sve takve veličine koje su od interesa za rad kontrolera i samog robota i koje se na pogodan način mogu pretvoriti u električne signale i dovesti putem fizičkih vodova do kontrolera nazivamo signalima iz okruženja /4/. Signali iz okruženja mogu da budu analogni i digitalni. Analogni signali se obrađuju u kontroleru preko A/D konvertora. Većina digitalnih signala su jednobitni (prekidački), npr. prekidački signali koji dolaze od okolnih uređaja ili signali koji se generišu promenom položaja prekidača na prednjoj ploči kontrolera, čime se omogućava operateru da jednostavno zadaje neke komande i kontrole u toku rada robota.

S druge strane, pojedini specifični sistemski podaci kontrolera su takve prirode da se mogu pogodno obraditi ukoliko se tretiraju kao U/I signali. Takve podatke nazivamo sistemskim U/I signalima. Tipični sistemski signali su npr. vrednosti sistemskih časovnika, koordinate vrha manipulatora, brzina vrhmanipulatora, itd. U sistemske U/I signale spadaju U/I podaci koji potiču sa hardverske op-

reme kontrolera /4/.

Očitavanje ulaznih signala, generisanje izlaznih signala i pristup sistemskim signalima iz robotskog jezika vrši se posredstvom posebne oblasti u memoriji (ulazno/izlazna oblast), koja sadrži trenutne vrednosti ulaznih i izlaznih signala (po 256 bajtova za ulazne, odnosno izlazne signale, koji su po konvenciji numerisani od 0 do 255). Korisnik može korišćenjem robotskog jezika da očitava podatke iz ulazne oblasti, kao i da očitava i menja podatke u izlaznoj oblasti. Sistemski programi U/I modula omogućuju da se istovremeno sa upisivanjem podataka u izlaznu oblast vrši generisanje odgovarajućih izlaznih signala.

Signali mogu da budu jednobitni i višebitni. Jednobitni signali su tipa LOG (logical), i mogu da imaju samo vrednosti 0 ili 1. Njihova vrednost se predstavlja u ulazno/izlaznoj oblasti kao jedan bit nekog bajta. U tom slučaju po pravilu i ostali bitovi tog bajta odgovaraju nekim jednobitnim signalima i kaže se da svi ti jednobitni signali pripadaju istoj grupi. Višebitni signali mogu da budu tipa BYTE, INTEGER, REAL ili POINT, zauzimaju respektivno jedan, dva, četiri ili četrnaest bajtova u ulazno/izlaznoj oblasti, respektivno, i počinju na granicama bajtova. Signali tipa POINT predstavljaju specifičan tip podataka u robotskom jeziku kontrolera, namenjenih za označavanje koordinata tačke u prostoru ili konfiguracije manipulatora. Šest polja od po dva bajta u podatku tipa POINT označavaju poziciju i orijentaciju u spoljašnjim ili unutrašnjim koordinatama, a sedmo predstavlja kontrolnu statusnu reč.

Karakterističan podatak za svaki signal je broj grupe, koji označava njegovu relativnu adresu u ulaznoj, odnosno izlaznoj oblasti (numeracija od 0 do 255). Za jednobitne signale karakterističan je i broj linije, odnosno redni broj bita unutar bajta koji predstavlja grupu jednobitnih signala (numeracija od 0 - bit najmanje težine, do 7 - bit najveće težine).

Svaki signal interno je opisan jednom strukturom podataka, koja ima oblik kao u sledećem primeru:

globalni naziv: SWITCH13

poreklo:	signal iz okruženja
smer:	ulazni
tip:	jednobitni
broj grupe:	15
broj linije:	3
port:	OF830H
pristup:	kod za vrstu pristupa

Globalno ime, ili globalni naziv signala, definisano je za sve sistemske signale. Globalni naziv za korisničke signale može, ali i ne mora da bude definisan. Podaci "pristup" i "port", koji su od značaja samo za signale iz okruženja, odnose se na vezu programa U/I modula i hardverske opreme kontrolera sa koje se očitavaju, odnosno na koju se šalju signali.

Broj i vrsta signala iz okruženja zavisi od radnog procesa i radne sredine u kojoj se robot koristi. Sistem omogućava korisniku da na pojedine priključke dovede različite signale, u skladu sa potrebama aplikacije i raspoloživom perifernom opremom. Kroz interaktivni postupak konfiguracije sistema, korisnik definiše spoljne signale linije koje su priključene na sistem u konkretnoj aplikaciji i značenje prekidača sa prednje ploče. Definisane karakteristika spoljašnjih signala, odnosno konfigurisanje U/I oblasti, vrši se memorisanjem skupa bitnih parametara za svaku od priključenih linija. Taj skup parametara se trajno memoriše u sistemu, i ne menja se sve do eventualne ponovne konfiguracije ulazno/izlazne oblasti, budući da sadrži one karakteristike koje se u toku korišćenja sistema ne menjaju (tip signala, broj grupe, U/I adresa, globalni naziv signala, itd.), ukoliko se ne vrše spoljašnja prevezivanja, uklanjanje pojedinih veza, dovodjenje novih signala, itd. Sve eventualne promene spoljašnjih veza i značenja pojedinih signala moraju se u sistemu ažurirati korišćenjem programa za konfiguraciju ulazno/izlazne oblasti pre početka korišćenja sistema u datom radnom procesu, dok je robot isključen.

Program za konfiguraciju ulazno/izlazne oblasti je interaktivni program koji omogućava da se u sistem dodaju novi opisi signala, da se obrišu ili izmene postojeći opisi signala, kao i da se izvrši listanje opisa signala.

### 3. PRISTUP SIGNALIMA IZ KORISNIČKIH PROGRAMA

U robotskim programima se spoljašnjim signalima pristupa radi očitavanja vrednosti ulaznih signala, ili radi generisanja izlaznih signala, kao i radi očitavanja vrednosti ranije generisanih izlaznih signala.

Umesto posebnih komandi za očitavanje/generisanje signala, odgovarajuće operacije automatski se izvršavaju pri nailasku na referencu na signal u robotskom programu. Generisanje izlaznih signala zadaje se na potpuno isti način kao i dodeljivanje vrednosti varijablama: ukoliko se referenca na izlazni signal nalazi sa leve strane komande za dodeljivanje vrednosti, vrši se generisanje odgovarajućeg izlaznog signala. Očitavanje vrednosti ulaznog ili izlaznog signala (bilo korisničkog signala iz okruženja, bilo sistemskog signala) vrši se kad god se referenca na signal nadje u nekom izrazu u robotskom programu.

Robotski programski jezik koji je ugrađen u kontroler omogućuje da se referisanje na signale u korisničkim programima ostvaruje na dva načina:

- navodjenjem tipa signala i njegovog broja grupe u ulazno/izlaznoj oblasti (direktan pristup signalima);
- korišćenjem simboličkog identifikatora koji je pridružen datom signalu (simboličko adresiranje signala).

Direktan pristup signalima izvodi se navodjenjem službene reči "EXT" (spoljašnji signal) ili "OUT" (izlazni signal), posle čega se za višebitne signale navodi tip podataka, a za jednobitne broj linije (celobrojna konstanta u opsegu /0,7/); referenca na signal završava se brojem grupe (celobrojna konstanta u opsegu /0,255/).

Primeri:

```
EXTGINTG14 ; ulazni 16-bitni signal
            ; na grupi linija 14
OUTGOG31   ; izlazni jednobitni signal
            ; na grupi 31, liniji 0
```

Simbolički pristup signalima vrši se navodjenjem simboličkog imena pridruženog signalu. Pritom se može koristiti i globalno ime (ugra-

djeno u sistem za sve systemske signale, odnosno opciono formirano pri konfigurisanju sistema), ili proizvoljno lokalno ime pridruženo signalu. Komanda za pridruživanje lokalnog imena signalu sadrži željeno lokalno ime, broj grupe za taj signal, oznaku da li je signal ulazni ili izlazni, i broj linije za jednobitne, odnosno oznaku tipa za višebitne signale.

Primeri:

```
OUT BIT: 466 ; deklaracija lokalnog
              ; naziva BIT za jednobitni
              ; signal na grupi 6,
              ; linija 4
          10
EXT ULAZ: BYTEG2 ; deklaracija lokalnog
                ; naziva ULAZ za višebitni
                ; signal tipa BYTE
                ; na grupi 2
OUT IZLAZ: INTG3 ; deklaracija lokalnog
                ; naziva IZLAZ za višebitni
                ; signal tipa INTEGER
                ; na grupi 3
```

### 4. KONCEPCIJA PRAĆENJA SPOLJNIH USLOVA

Jedan od bitnih preduslova za robotizaciju složenih zadataka je mogućnost robota da autonomno i pravovremeno reaguje na promene u okruženju. U tom cilju, neophodno je da robotski kontroler pruži mogućnost praćenja spoljnih uslova paralelno sa obavljanjem osnovnog manipulacionog zadatka i preduzimanja odgovarajuće akcije pri nailasku relevantnog spoljnog asinhronog događaja.

U/I modul kontrolera opšte namene pruža mogućnost praćenja spoljnih uslova zadatih u vidu prostih relacionih izraza preko kojih se stavljaju u neki odnos ulazni signal, s jedne strane, i neka konstanta, varijabla ili neki drugi signal (ulazni ili izlazni) s druge strane. Za razliku od običnih relacionih izraza, koji se ispituju samo onda kada se na njih naidje u toku interpretacije programa, ovakvi spoljni uslovi se moraju najpre deklarirati u robotskom programu, a kada se potom predje na njihovo ispitivanje (korišćenjem posebne robotske komande), ono se periodično obavlja sve dok se ne ustanovi da je ispitivani uslov zadovoljan. Ispitivanje se obavlja potpuno asinhrono u odnosu na rad interpreta-

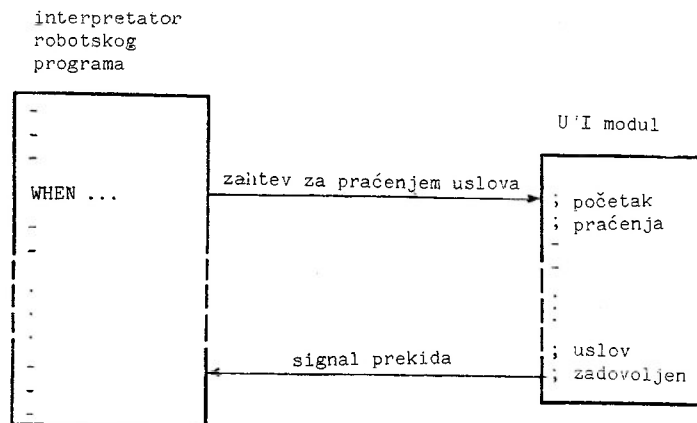
tora robotskih komandi.

Da bi se omogućilo referisanje na neki spoljašnji uslov u robotskom programu i njegovo praćenje, potrebno je najpre deklarirati taj uslov u programu korišćenjem komande COND. Deklaracija uslova dodeljuje uslovu simbolički naziv i opisuje relacioni izraz koji se ispituje pri kasnijem praćenju uslova. Taj relacioni izraz mora obavezno da sadrži referencu na neki ulazni signal (korisnički ili sistemski), referencu na podatak koji se poredi sa spoljašnjim signalom i željeni tip poredjenja (relacioni operator). Podatak koji se poredi sa ulaznim signalom može da bude varijabla, konstanta ili neki drugi signal (ulazni ili izlazni).

Primeri:

```
COND USPEH: A > 3.5 ;A je ranije definisan
                ;kao realni signal
COND NOVOST: X <= Y ;poredjenje signala X
                ;sa promenljivom Y
```

Praćenje ovako deklarisanog spoljašnjeg uslova započinje onda kada se u programu nađe odgovarajuća komanda (komanda WHEN, odeljak 5), unutar koje se zadaju prioritet uslova, potrebna učestanost ispitivanja i tip praćenja uslova. Praćenje se zatim odvija asinhrono, nezavisno od interpretacije robotskog programa, kao što to ilustruje sl.1, sve dok se ne ustanovi da je uslov zadovoljen. Tada U/I modul šalje interpretatoru zahtev za prekidanjem robotskog programa.



Sl.1. - Asinhrono praćenje spoljašnjeg uslova

Svakom uslovu čije se praćenje zahteva, korisnik može po želji da dodeli neki prioritet. Ukoliko je prioritet zadovoljenog uslova veći od prioriteta procesa koji trenutno obavlja interpretator, interpretator prekida započeti proces i prelazi na procesiranje zadovoljenog uslova. Procesiranje zadovoljenog uslova podrazumeva preduzimanje odgovarajućih akcija definisanih komandom za početak praćenja uslova, pre nego što se nastavi prekinuti proces. Ukoliko prioritet zadovoljenog uslova nije veći od prioriteta procesa koji obavlja interpretator u trenutku kada dodje do zadovoljenja uslova, zadovoljeni uslov se procesira

tek onda kada interpretator obavi sve prioritetnije poslove.

Ako interpretator obavlja proces relativno visokog prioriteta i za to vreme više ispitivanih uslova nižeg prioriteta postane zadovoljeno, rad interpretatora se ne prekida, a zadovoljeni uslovi se smeštaju u prioritetni red čekanja. Nakon završetka obavljanog procesa, interpretator počinje sa procesiranjem najprioritetnijeg uslova iz reda.

Asinhrono ispitivanje spoljašnjih uslova može se vršiti periodično u svakom osnovnom inter-

valu (10 ms), u svakom desetom osnovnom intervalu, ili u svakom stotom osnovnom intervalu. Korisnik to po volji specificira u komandi WHEN, navodjenjem jedne od tri službene reči: FAST (ispitivanje u svakom intervalu, odnosno sa kratkom periodom ispitivanja), MED (ispitivanje u svakom desetom intervalu, odnosno sa srednjom periodom ispitivanja) i SLOW (ispitivanje u svakom stotom intervalu, odnosno sa drugom periodom ispitivanja).

Praćenje uslova u većini slučajeva može da se prekine čim se ustanovi da je uslov zadovoljen. Ovo je ujedno i standardni način praćenja spoljnih signala od strane U/I modula i označava se kao "praćenje ivice" (praćenje tipa EDGE). Termin potiče od toga što se zadovoljenje nekog uslova može shvatiti kao specifični softverski prekidni "impuls" koji U/I modul šalje interpretatoru u trenutku kada ga obavestava o tome da je uslov zadovoljen, i da već samim nailaskom "prednje ivice" tog "impulsa" interpretator dobija trajan zahtev za prekidom.

Moguće je zamisliti i situaciju kada neki zadovoljeni uslov, koji interpretator nije još procesirao zbog niskog prioriteta, više nije zadovoljen u trenutku kada interpretator može da predje na njegovu obradu. Takva situacija nastupa, npr., u slučaju kada je uslov definisan činjenicom da je neki radni predmet prisutan na pokretnoj traci, a robot ga nije na vreme preneo sa nje. U međuvremenu, traka se pomerila i uslov više nije ispunjen. Očigledno da u takvim situacijama nije dovoljno da praćenje uslova prestane čim se prvi put ustanovi da je uslov zadovoljen. Ukoliko se odmah po pristizanju informacije o zadovoljenju uslova nije prešlo na njegovo procesiranje, treba nastaviti sa praćenjem tog uslova. Ovakvo praćenje se označava kao "praćenje nivoa" (praćenje tipa LEVEL), s obzirom da "prekidni impuls" mora da "traje" do trenutka kada se predje na procesiranje uslova.

## 2. OBRADA SPOLJNIH USLOVA U KORISNIČKIM PROGRAMIMA

Praćenje spoljašnjih uslova reguliše se u korisničkim programima korišćenjem komandi WHEN, ENABLE, DISABLE I IGNORE.

Komanda WHEN sadrži zahtev za praćenje nekog spoljašnjeg uslova, opisuje način praćenja

uslova, određuje prioritet uslova i definicije akcije koju treba preduzeti pri procesiranju uslova. Sintaksa ove komande ima opšti oblik:

WHEN pri frek tip uslov DO akcija

gde parametar "uslov" označava simbolički naziv uslova definisan prethodno odgovarajućom deklaracijom, "akcija" označava opis akcija koje sistem obavlja kada se uslov zadovolji (robotska komanda, npr. dodeljivanje vrednosti, CALL, itd.), a "tip" određuje da li se zahteva "praćenje ivice" ili "praćenje nivoa". Prioritet uslova specificira se parametrom "pri" (sadrži proizvoljnu celobrojnu konstantu u opsegu /1,15/, pri čemu većoj vrednosti odgovara viši prioritet), a željena frekvencija staniranja parametrom "frek" (može da ima vrednosti FAST, MED, ili SLOW). Neki od ovih parametara ne moraju se navesti, i tada se podrazumeva da uzimaju standardne vrednosti, kao što se to vidi iz primera:

WHEN 4; FAST LEVEL NOVOST DO CALL A

; testiranje uslova NOVOST uz "praćenje nivoa",  
; sa frekvencijom FAST i prioritetom 4;  
; po zadovoljenju uslova,  
; skok na rutinu prekida A,

WHEN USPEH DO OUTSIG = YES

; automatski dodeljen prioritet 1,  
; "praćenje ivice" i frekvencija MED

Praćenje spoljašnjeg uslova prestaje u momentu kada dati uslov postane zadovoljen i pritom izazove prekid rada programa. Ukoliko treba sprečiti da zadovoljenje nekog uslova izazove prekid rada programa, a da se pri tome ipak sačuva informacija da li je uslov zadovoljen ili ne, zbog eventualnog kasnijeg procesiranja istog uslova, koristi se komanda DISABLE. Drugim rečima, ukoliko se neki uslov prati, i ukoliko se naidje na komandu DISABLE koja se odnosi na taj uslov, sistem nastavlja sa praćenjem uslova sve dok uslov ne postane zadovoljen, ukoliko se radi o "praćenju ivice", odnosno i nakon toga, ukoliko se radi o "praćenju nivoa". Zadovoljeni uslov se stavlja u red čekanja, u skladu sa svojim prioritetom, ali ne može da izazove prekid rada programa i da bude procesiran. Procesiranje tog uslo-

va moći će da se obavi tek kada se naidje na odgovarajuću komandu ENABLE. Način funkcionisanja komandi ENABLE i DISABLE vidi se iz sledećih primera:

```
DISABLE INCOND ;onemogućavanje prekida izazvanog
                zadovoljenjem uslova INCOND
DISABLE                ;onemogućavanje svih prekida
ENABLE INCOND ;ponovno omogućavanje
                ;delovanje uslova INCOND
DISABLE 2                ;onemogućavanje prekida
                ;sa prioritetom 1 i 2
ENABLE 13                ;omogućavanje prekida
                ;sa prioritetom 13,14 i 15
```

Komanda IGNORE je slična komandi DISABLE u pogledu sintakse, s tim što se uslov na koji je primenjena komanda IGNORE više ne prati, i da bi se ponovo omogućilo njegovo praćenje potrebno je to ponovo zahtevati odgovarajućom komandom WHEN. Ukoliko je uslov bio zadovoljen pre nailaska komande IGNORE, a još nije bio procesiran zbog niskog prioriteta, komandom IGNORE biva uklonjen iz reda čekanja, čime se trajno onemogućava da bude obradjen. Kao i komanda DISABLE, komanda IGNORE može da deluje na jedan uslov, na grupu uslova sa prioritetom manjim ili jednakim navedenom prioritetu, ili na sve uslove koji se prate:

```
IGNORE CONDI ;prekid praćenja uslova CONDI
IGNORE 2                ;prekid praćenja uslova
                ;s prioritetom 1 i 2
IGNORE                ;prekid praćenja svih
                spoljašnjih uslova
```

## 6. DISKUSIJA

U/I modul robotskog kontrolera opšte namene u celini je implementiran na programskom jeziku PL/M. Programi ovog modula mogu se okvirno podeliti u tri grupe:

a/ Programi za neposredno očitavanje/generisanje U/I signala.

Ovo su programi koji omogućavaju očitavanje ulaznih signala sa raznih U/I portova i njihovo memorisanje u U/I oblasti, generisanje izlaznih signala i njihovo slanje prema okruženju, kao i obradu sistemskih signala. U ovu grupu spadaju programi kako za sinhronu tako i za asinhronu obradu U/I signala (praćenje spoljnih signala i uslova), i svi oni vode ra-

čuna o fizičkim adresama pojedinih U/I signala.

b/ Programi za pristup signalima iz robotskih programa. Ovu grupu programa praktično čine programi preko kojih se ostvaruju pojedine komande robotskog programskog jezika RL (COND, WHEN, IGNORE, ...).

c/ Programi vezani za softversku konfiguraciju U/I modula. Programi iz ove grupe čine podmodul U/I modula koji se ne koristi u realnom vremenu, već se poziva iz monitora sistema /4/, u cilju izvodjenja (re)konfiguracije modula, na način opisan u odeljku 2.

Sistem u ovom trenutku ne poseduje mogućnost obrade informacija sa senzora, ali se planira njegovo proširenje u tom smislu. Asinhrona obrada podataka sa senzora sasvim se uklapa u koncepciju asinhronog praćenja spoljnih uslova, kako se to može videti iz /6/. Senzori, npr. često imaju veliku ulogu pri ostvarivanju tzv. praćenog kretanja (guarded motion) /7/, koje je potpuno podržano asinhronom obradom spoljnih uslova /3/. Spoljni uslov se, naime, može veoma prirodno zadati npr. u obliku:

```
COND BREAK: SEN1 > FORCE1
```

što bi moglo da predstavlja uslov da podatak o sili koja deluje na radni predmet prilikom njegovog spajanja sa drugim predmetom, a koji pruža senzor sile SEN1, ne sme da predje prag FORCE1, inače može da dodje do suvišnog naprezanja predmeta. Ovo je tipičan slučaj praćenog kretanja, koje se vrlo često sreće u robotskim zadacima.

## 7. ZAKLJUČAK

Opisani U/I modul robotskog kontrolera opšte namene, realizovanog u Institutu "Mihailo Pupin", u potpunosti podržava i sinhrono i asinhrono procesiranje spoljnih signala i uslova. U kontroler je ugradjen robotski programski jezik, RL, koji omogućuje korisnicima da programiraju veliki broj zadataka koje robot treba da obavi. U robotskim programima se može zahtevati praćenje spoljnih signala i uslova (asinhrono procesiranje, paralelno sa izvršavanjem manipulacionog zadatka), uz preduzimanje odgovarajućih akcija u slučaju zadovoljenja uslova, kao i neposredno očitavanje/gene-

risanje spoljnih signala (sinhrono procesiranje). U/I modul pruža mogućnost obrade pojedinih sistemskih podataka na potpuno isti način kao i U/I signala.

U sistem je ugrađen skup programa koji omogućavaju interaktivnu softversku rekonfiguraciju U/I modula u cilju prilagođavanja promena u okruženju ili u aplikaciji. U sistemu se

planira značajno proširenje dodavanjem senzora. Obradjene informacije sa senzora moći će da se procesiraju preko U/I modula kao i drugi spoljni signali i uslovi. U/I modul kontrolera koncipiran je tako da pruža povoljne predulove za primenu kontrolera u hijerarhijskom sistemu upravljanja fleksibilnom proizvodnom ćelijom.

## 8. LITERATURA

- /1/ Vukobratović, M., Kirčanski, N., Stokić, D., Kirčanski, M., Karan, B.: General Purpose Controller for Industrial Manipulators, Proc. of the Second Yugoslav-Soviet Symp. on Applied Robotics, Arandjelovac, YU, 1984., pp. 1-15.
- /2/ Vukobratović, M., Karan, B., Kirčanski, M., Kirčanski, N.: Koncept multiprocesorskog robotskog kontrolera, Zbornik radova sa XXX konferencije ETAN-a, Herceg Novi, 1986., Sveska XI, str. 143-148.
- /3/ Vukobratović, M. i ost.: Uvod u robotiku, Institut Mihajlo Pupin, Beograd, 1986., str. 241-266
- /4/ Karan, B., Timotijević, M., Djurović, M., Đevdžić, V., Josifović, N.: A System for Programming the UMS-7 Industrial Robot, Proc. of 3rd Soviet-Yugoslav Symp. on Applied Robotics, Moscow, USSR, Jun 9-11, 1986., pp. 35-43.
- /5/ Karan, B.: A Software System for Teaching and Commanding the Industrial Robots, Proc. of IFAC Symposium on Robot Control, Barcelona, Spain, Nov. 6-8., 1985., pp. 51-56.
- /6/ Milovanović, R.: An Analysis of General Purpose Robot Programming Language, Proc. of IFAC Symposium on Robot Control, Barcelona, Spain, Nov. 6-8., 1985., pp. 45-50
- /7/ Lozano-Perez, T.: Robot Programming, Proc. of IEEE, Vol. 71, No. 7, 1983., pp. 821-841.
- /8/ Taylor R.H., Summers P.D., Meyer J.M.: AML: A Manufacturing Language, Robotics Research, Vol. 1, No. 3, 1982., pp. 19-41.
- /9/ Unimation Inc.: User's Guide to VAL: a Robot Programming and Control System, Version 12, 1980., Danbury.