



Mitrović, S.¹⁾, Jakovljević, Ž.²⁾

DISTRIBUIRANO UPRAVLJANJE PNEUMATSKIM ROBOTOM BAZIRANO NA IEC 61499 I 802.15.4 STANDARDIMA^{3) 4)}

Rezime

Potrebe današnjeg tržišta nameću primenu i razvoj novih tehnologija koje omogućavaju efikasno generisanje malih proizvodnih serija uz ekstremno brzu reakciju prilikom pojave zahteva za novim proizvodima. Proizvodnja mora biti fleksibilna, sa proizvodnim sredstvima koja poseduju rekonfigurabilnu hardversku i softversku strukturu. Decentralizacijom upravljanja i distribucijom upravljačkih zadataka na više inteligentnih uređaja pojednostavljuje se konfigurisanje i rekonfigurisanje automatskih proizvodnih sistema, čime se mogu ostvariti značajne uštede u energiji, materijalu i vremenu. U okviru ovog rada opisan je distribuirani sistem upravljanja pneumatskim robotom za bojenje, zasnovan na standardu IEC 61499, uz upotrebu IEC 802.15.4 bežičnog komunikacionog protokola.

Ključne reči: distribuirano upravljanje, IEC 61499, IEC 802.15.4, ZigBee, pneumatski robot.

1. UVOD

Inicijativa "Industrie 4.0", koja podrazumeva visokofleksibilnu, individualizovanu i efikasnu proizvodnju, predstavlja jedan od glavnih pokretača napredne digitalizacije proizvodnje kroz primenu koncepata kibernetско-fizičkih sistema (CPS – *engl. Cyber Physical Systems*) i interneta stvari (IoT – *engl. Internet of Things*) [1]. Fizički objekti (pametni uređaji – senzori i aktuatori) povezuju se u funkcionalne celine pomoću odgovarajućih digitalnih modela, a posredstvom interneta i interneta stvari, čime se dolazi do kibernetско-fizičkih proizvodnih sistema [2]. Osnovni gradivni elementi pomenutih sistema – pametni (*engl. smart*) uređaji, podržavaju modularni pristup gradnji, kao i opštu vertikalnu i horizontalnu umreženost, portabilni su i interoperabilni, što omogućava izvedbe koje se lako i brzo mogu rekonfigurirati prema novonastalim zahtevima [3].

Predviđeno je da se upravljački zadaci distribuiraju na pametne uređaje, pri čemu se saopšteni zadatak izvršava kroz međusobnu komunikaciju, gde svaki od pomenutih uređaja izvršava samo jedan segment saopštenog zadatka. Na opisan način dolazi se do rešenja koje je moguće lako modifikovati prema zahtevima, što značajno utiče na uštede u pogledu energije, vremena i materijala [4]. Tehnologija, odgovarajuća tehnička sredstva, kao i prateći standardi neophodni za adekvatnu primenu pomenutog pristupa još su u fazi razvoja, te se za potrebe istraživanja koriste dostupne, univerzalne računarske platforme, najčešće na bazi mikrokontrolera i mikroracunara.

U okviru ovog rada navedeni koncepti biće opisani kroz analizu razvijenog distribuiranog sistema upravljanja primenjenog na pneumatskom robotu sa dva translatorna i jednim rotacionim stepenom slobode kretanja i hvatačem kao završnim uređajem, pri čemu je programirana operacija bojenja delova potapanjem u posudu sa bojom.

Za distribuciju i simulaciju upravljačkog zadatka korišćeno je HOLOBLOC FBDK [5] (*engl. Function Block Development Kit*) softversko razvojno okruženje, bazirano na standardu IEC 61499 (Function Block) kojim je definisana arhitektura distribuiranog sistema upravljanja. Razvijeni upravljački softver, opisan

¹⁾ Mast. inž. maš. Stefan Mitrović, Lola institut, Beograd, stefan.mitrovic@li.rs

²⁾ Prof. dr Živana Jakovljević, Univerzitet u Beogradu, Mašinski fakultet, zjakovljevic@mas.bg.ac.rs

³⁾ U okviru ovog rada saopštavaju se rezultati istraživanja koja se sprovode na projektima TR35004, TR35020 i TR35023 koje finansijски podržava MPNTR Republike Srbije.

⁴⁾ U okviru ovog rada korišćene su komponente dobijene donacijom kompanije Predstavništvo SMC Industrial Automation SRB Laboratoriji za automatizaciju proizvodnih procesa. Ovom prilikom autori izražavaju zahvalnost kompaniji Predstavništvo SMC Industrial Automation SRB.

funkcijskim blokovima, u nastavku je preveden u odgovarajući mikrokontrolerski kôd koji je primenjen na dostupnim mikrokontrolerima i testiran na laboratorijskom pneumatskom robotu.

Ovaj rad ima sledeću strukturu. U drugom odeljku izvršen je opis hardverske strukture i zadatka upravljanja pneumatskim robotom. Treći odeljak sadrži opis razvijenih funkcijskih blokova sa posebnim osvrtom na način njihove primene i programsku logiku. U četvrtom odeljku detaljno se opisuje upotrebljena mikrokontrolerska platforma, modul za bežičnu komunikaciju i način povezivanja sistema u celinu. Eksperimentalna verifikacija i rezultati izloženi su u petom odeljku. Konačno, u šestom odeljku date su zaključne napomene i smernice za dalja istraživanja.

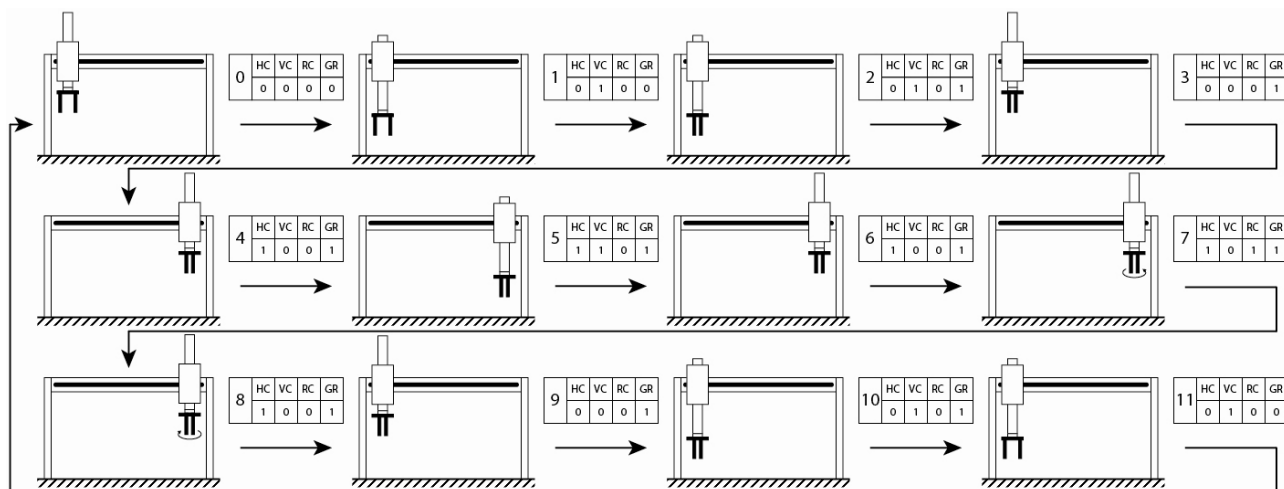
2. OPIS SISTEMA I UPRAVLJAČKOG ZADATKA

Pomenuti „pick and place“ robot poseduje četiri inteligentna aktuatora - tri translatorna i jedan rotacioni pneumatski cilindar. Od tri translatorna pneumatska cilindra, jedan jednosmernog dejstva služi za otvaranje i zatvaranje hvatača – završnog uređaja. Preostala dva translatorna cilindra pokreću završni uređaj u horizontalnom i vertikalnom pravcu. Rotacioni cilindar ostvaruje zakretanje završnog uređaja oko vertikalne ose u opsegu od $\pm 180^\circ$. U odnosu na to kako su orijentisani na robotu, inteligentni aktuatori su u šemama obeleženi sa: (1) „HC“ – horizontalni cilindar, obavlja horizontalnu translaciju završnog uređaja; (2) „VC“ – vertikalni cilindar, obavlja vertikalnu translaciju završnog uređaja; (3) „RC“ – rotacioni cilindar, obavlja zakretanje završnog uređaja oko vertikalne ose i (4) „GR“ (*engl. Gripper*), pneumatski cilindar koji otvara i zatvara hvatač – završni uređaj. Od navedenih aktuatora, HC, VC i RC pored samog pneumatskog cilindra poseduju po jedan električno upravljani bistabilni razvodnik 5/2 za upravljanje radom cilindra i dva krajnja prekidača. GR aktuator ima monostabilni razvodnik 5/2 za upravljanje radom cilindra i ne poseduje krajnje prekidače zbog malih dimenzija. Svaki od navedenih aktuatora opremljen je sopstvenim upravljačkim sistemom – čvorom, koji se sastoji od mikrokontrolera i komunikacionog modula kojim se vrši bežično povezivanje sa preostala tri čvora.

Pneumatski robot ima za zadatak da spusti hvatač, uhvati i podigne objekat, a zatim, horizontalnim translatornim kretanjem, locira objekat iznad posude sa bojom. Sledi potapanje objekta u posudu, bez otpuštanja. Po podizanju objekta iz posude, sledi otesanje viška boje sa dve brze, uzastopne rotacije (+180, -180), praćeno povratkom u prvobitnu poziciju u horizontalnom pravcu, spuštanjem i otpuštanjem obojenog objekta iz hvatača. Nakon otpuštanja objekta, robot se sa otvorenim hvatačem vraća u inicijalnu poziciju, spreman da ponovi opisani ciklus sa pojavom novog objekta. Rad robota opisan je sekvencom:

$$VC+GR+VC-HC+VC+VC-RC+RC-HC-VC+GR-VC-, (1)$$

pri čemu se smatra da „+“ podrazumeva izvlačenje klipnjače, odnosno, rotaciju u matematički negativnom smeru, a „-“ podrazumeva uvlačenje klipnjače, odnosno, rotaciju u matematički pozitivnom smeru. Na slici 1 dat je šematski prikaz opisanog sistema.

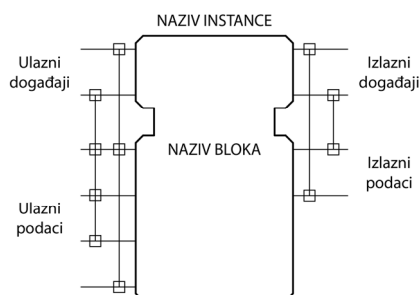


Slika 1. Šematski prikaz rada pneumatskog robota – sekvence.

3. FUNKCIJSKI BLOKOVI

IEC 61499 je standard koji definiše razvoj, primenu i ponovnu upotrebu funkcijskih blokova u okviru industrijske automatizacije [6]. U okviru IEC 61499 standarda, programiranje ponašanja sistema vrši se funkcijskim blokovima, enkapsuliranim programskim jedinicama kojima se mogu modelirati kako hardverske, tako i softverske komponente posmatranog sistema [7]. Funkcijski blokovi su zamišljeni kao mogući odgovor na sve izraženiji problem programiranja složenih zadataka u automatizaciji, gde se često pojavljuju obimni kodovi koje je teško, ili nemoguće modifikovati i/ili ponovo primeniti u novoj postavci sistema. Posledično, dolazi do velikih gubitaka u vremenu uzrokovanih ponovnom izradom softvera za već postojeći hardver. Upotrebom funkcijskih blokova mogu se eliminisati navedeni nedostaci, čime se znatno olakšava rekonfiguracija sistema.

IEC 61499 može se posmatrati kao standard koji definiše objektno orijentisani grafički jezik pomoću kog je moguće programiranje centralizovanih i distribuiranih sistema upravljanja. Funkcijski blokovi analogni su klasama, apstraktnim modelima čije instance – objekti u konkretnim programima obavljaju namenjenu funkciju. Opšti model funkcijskog bloka dat je na slici 2.

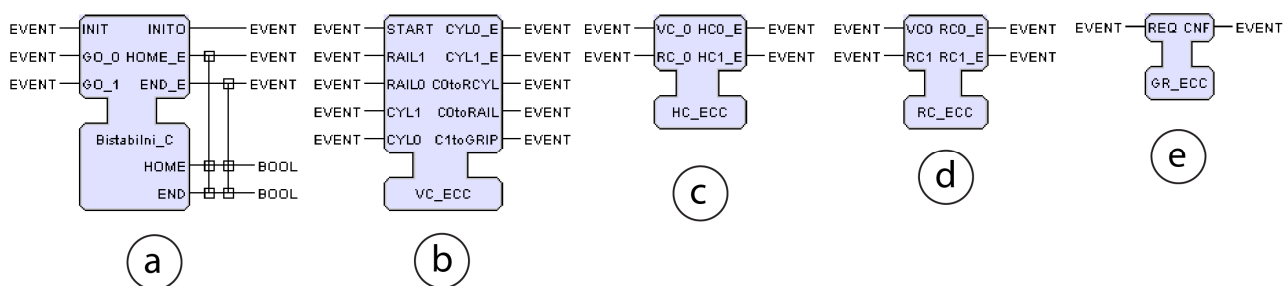


Slika 2. Opšti model funkcijskog bloka.

Kod funkcijskog bloka, gornji deo naziva se glava, a donji se naziva telo. Za glavu bloka vezuju se ulazni (levo) i izlazni (desno) događaji, dok se za telo vezuju ulazni (levo) i izlazni (desno) podaci. Blok započinje sa obavljanjem dodeljenog zadatka po prispeću odgovarajućeg događaja. Nakon obavljenog zadatka, generiše se odgovarajući izlazni događaj koji se šalje dalje, kao ulazni događaj drugih blokova – komponenti sistema. Zadatak koji blok obavlja definisan je u okviru funkcija unutar bloka. Funkcije se aktiviraju ulaznim događajima, izvršavaju se nad ulaznim promenljivim i generišu izlazne promenljive i izlazne događaje.

Tokom modeliranja ponašanja sistema javlja se potreba za različitim funkcijskim blokovima, jednostavnim ili složenim, koji obavljaju različite zadatke. U zavisnosti od okruženja u kom se vrši programiranje, biće dostupan različit broj već gotovih blokova, spremnih za upotrebu u programu jednostavnim pozivanjem iz biblioteke. Takođe, korisnik može razviti sopstvene, specijalne funkcijske blokove u slučaju da se javi potreba.

U našem slučaju, za modeliranje ponašanja sistema korišćeno je razvojno okruženje FBDK. Unutar okruženja, prisutna je biblioteka najčešće korišćenih, tipskih funkcijskih blokova, čija upotreba prilikom izrade aplikacija olakšava i ubrzava celokupan proces. Pored dostupnih, bilo je potrebno razviti još 5 novih blokova radi ostvarivanja željene funkcije celokupnog sistema. Svi novi blokovi prikazani su na slici 3.

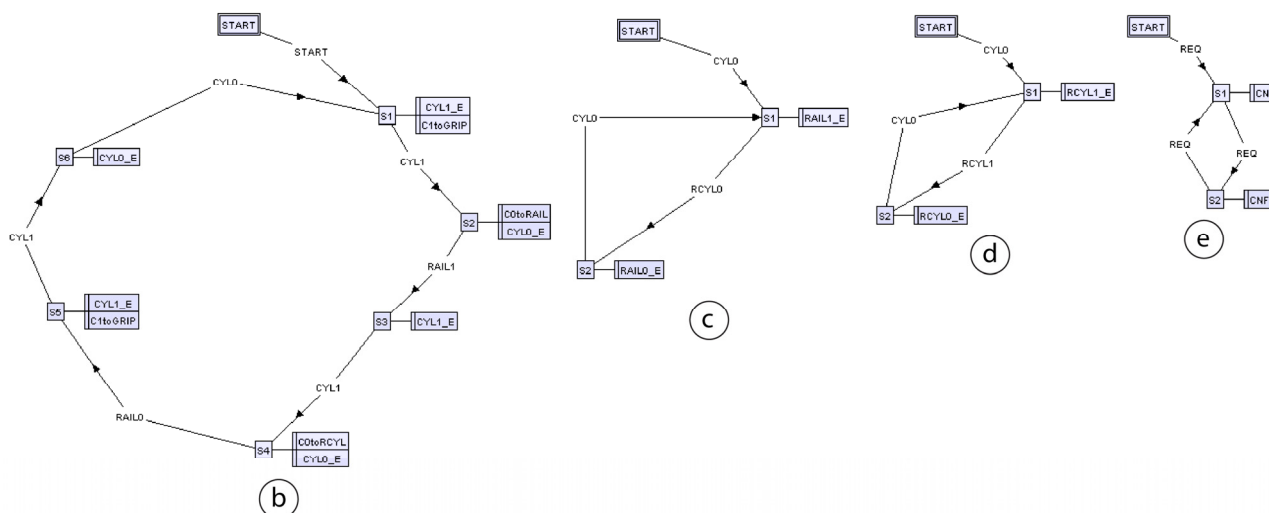


Slika 3. Razvijeni funkcijski blokovi.

Na slici 3.a nalazi se prikaz interfejsa funkcijskog bloka „Bistabilni_C“ kojim se modelira ponašanje dvosmernog pneumatskog cilindra. Ulazni događaj „INIT“ postavlja blok u inicijalno stanje, sa uvučenom klipnjačom, pri čemu se po izvršenoj inicijalizaciji generiše izlazni događaj „INITO“. Ulazni događaj „GO_0“ saopštava bloku-cilindru da uvuče klipnjaču u slučaju da je izvučena i da po uvlačenju klipnjače generiše odgovarajući izlazni događaj „HOME_E“, uz promenu vrednosti izlazne promenljive „HOME“ na 1, a „END“ na 0. U slučaju da je klipnjača već uvučena, „GO_0“ događaj biće prihvaćen, ali nikakvih događaja neće biti na izlazu, niti će biti menjane vrednosti izlaznih promenljivih. Slično važi i za ulazni događaj „GO_1“ kojim se saopštava da treba izvući klipnjaču cilindra u krajnji položaj, generisati odgovarajući izlazni događaj „END_E“ i postaviti vrednosti izlaznih promenljivih – „HOME“ na 0, a „END“

na 1. U slučaju da je klipnjača već izvučena, po pristizanju „GO_1“ događaja, neće biti nikakvih promena na izlazu u pogledu događaja i vrednosti promenljivih. Bitno je napomenuti da su za funkcionisanje simulacije neophodni samo događaji, ali se zbog generisanja odgovarajućih grafičkih elemenata moraju koristiti izlazne promenljive u bloku „Bistabilni_C“.

Na slici 3.b prikazan je interfejs funkcijskog bloka „VC_ECC“ kojim se upravlja vertikalnim pneumatskim cilindrom unutar aplikacije. Blok „VC_ECC“ sadrži svu upravljačku logiku neophodnu za ispravan rad vertikalnog pneumatskog cilindra. Logika je saopštena u vidu ECC dijagrama (engl. Execution Control Chart) koji je vrlo sličan SFC (engl. Sequential Function Chart) grafičkom jeziku za programiranje programabilnih kontrolera. ECC upravljačka logika primenjena unutar bloka „VC_ECC“ predstavljena je na slici 4.b.



Slika 4. ECC dijagrami svih upravljačkih funkcijskih blokova.

„VC_ECC“ blok nema ulaznih, ni izlaznih promenljivih – blok operiše samo sa događajima. Na ulazu se očekuju događaji bitni za rad – „START“ (pritisnut start taster kojim započinje rad sistema), „RAIL1“ i „RAILO“ (horizontalni cilindar izvučen i horizontalni cilindar uvučen), a prati se i stanje samog vertikalnog cilindra sa „CYL1“ i „CYL0“ (vertikalni cilindar izvučen i vertikalni cilindar uvučen). U zavisnosti od ulaznog događaja i ECC stanja (S1, S2, S3...) u kom se trenutno nalazi funkcijski blok „VC_ECC“, vertikalni cilindar će izvlačiti ili uvlačiti klipnjaču, ili mirovati, a o promenama stanja generisaće se odgovarajući izlazni događaji „CYL0_E“ i „CYL1_E“ – vertikalni cilindar uvučen i vertikalni cilindar izvučen, koji će se koristiti za simulaciju, kao i događaji koji ukazuju na to da je potrebno odgovarajuću informaciju proslediti drugim resursima u sistemu kako bi mogli da nastave sa odgovarajućim akcijama i to: (1) „C0toRCYL“ – poslati informaciju rotacionom cilindru, (2) „C0toRAIL“ – poslati informaciju horizontalnom cilindru i (3) „C1toGRIP“ – poslati informaciju hvataču. Navedene informacije se prihvataju u funkcijskim blokovima koji odgovaraju pojedinim resursima i omogućuju kretanje kroz njihove pripadajuće ECC dijagrame.

Funkcijski blokovi slične strukture i logike korišćeni su i kod preostalih aktuatora, pri čemu razlike nastaju zbog različitih ulaznih događaja koji izazivaju promene stanja. Na slici 3.c prikazan je „HC_ECC“ blok koji predstavlja virtuelnu upravljačku jedinicu horizontalnog cilindra. Integrisana logika opisana ECC dijagramom data je na slici 4.c. Promene stanja horizontalnog cilindra izazivaju samo dva ulazna događaja – „VC0“ (vertikalni cilindar uvučen) i „RC0“ (rotacioni cilindar u početnom položaju). Izlazni događaji kojima se upravlja cilindrom su „HC0_E“ (horizontalni cilindar uvučen) i „HC1_E“ (horizontalni cilindar izvučen).

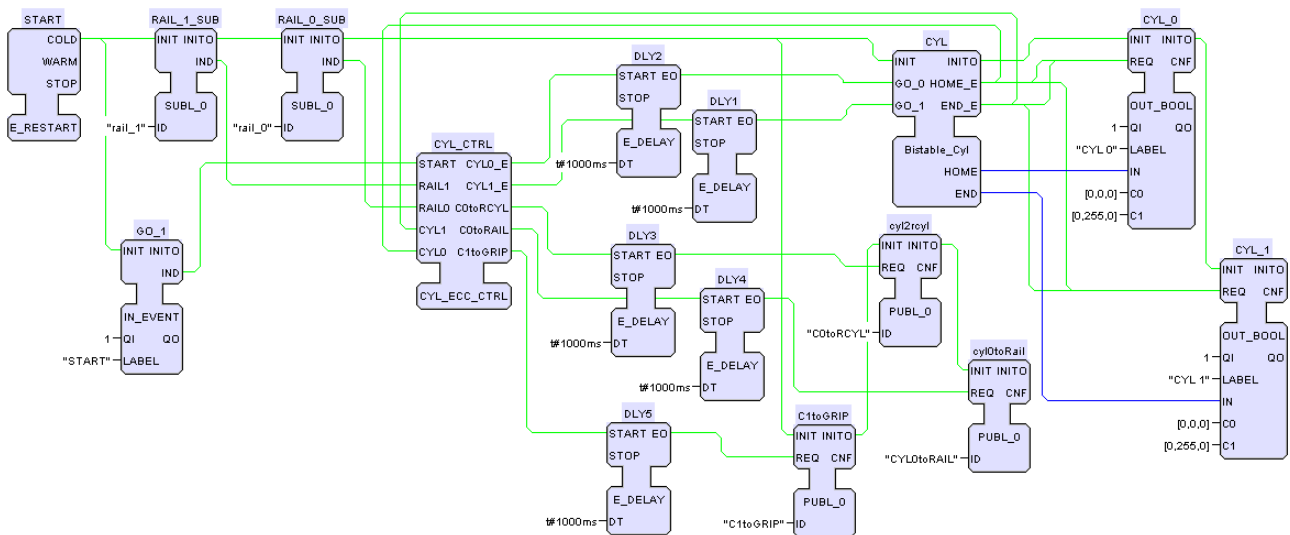
Na slici 3.d prikazan je „RC_ECC“ blok, kojim se upravlja rotacionim cilindrom. Odgovarajući ECC dijagram dat je na slici 4.d. Rotacioni cilindar se pomera u zavisnosti od stanja vertikalnog cilindra i sopstvenog stanja, te se na ulazu prate događaji „VC0“ (vertikalni cilindar uvučen) i „RC1“ (rotacioni cilindar u krajnjem položaju).

Na slici 3.e prikazan je „GR_ECC“ funkcijski blok kojim se upravlja otvaranjem i zatvaranjem hvatača robota. Prateći ECC dijagram dat je na slici 4.e. S obzirom na to da stanje hvatača zavisi isključivo od stanja vertikalnog cilindra, postoji samo jedan ulazni događaj – „VC1“ (vertikalni cilindar izvučen), dok se na izlazu generiše odgovarajući izlazni događaj koji u zavisnosti od trenutnog stanja vrši otvaranje, odnosno, zatvaranje hvatača.

Iako upravljanje svakog od aktuatora prati samo određene događaje, ponavljanje pomenutih događaja

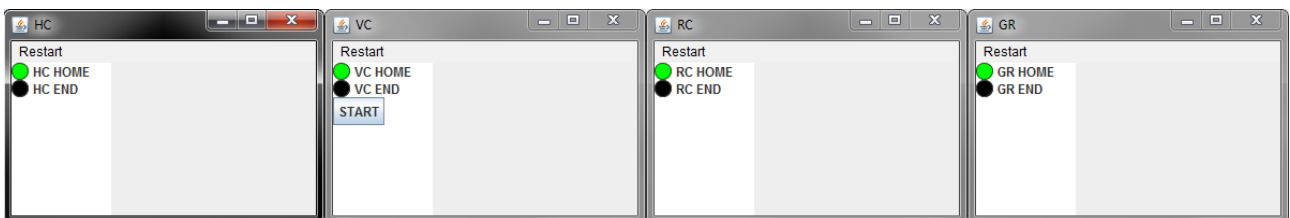
tokom jednog ciklusa nije isključeno. Međutim, ECC logika svakog od aktuatora dozvoljava rad samo ako se odgovarajuće stanje u kom je aktuator poklapa sa prispelim ulaznim događajima, čime je otklonjena mogućnost neželjenog pokretanja izvršnih organa.

Povezivanjem blokova u funkcionalnu celinu kreira se aplikacija na način kao da će se upravljanje vršiti centralizovano. U okviru aplikacije, raspoloživi funkcijski blokovi istanciraju se pod određenim imenom, a zatim se povezuju. Zelene linije predstavljaju veze koje prenose događaje, a plave linije predstavljaju veze koje prenose podatke. Dodeljivanjem delova aplikacije pojedinim resursima (u ovom slučaju inteligentnim aktuatorima), vrši se distribucija upravljačkog zadatka. Tokom pomenutog procesa pojedine veze između funkcijskih blokova se prekidaju i potrebno ih je ponovo uspostaviti uvođenjem komunikacije reprezentovane komunikacionim funkcijskim blokovima. Pošto je upravljanje razmatranim robotom vođeno događajima, između pojedinih resursa neophodna je samo razmena informacija o događajima, pa su upotrebljeni *publish* (PUBL_0) i *subscribe* (SUBL_0) blokovi koji omogućuju razmenu informacija navedenog tipa. S obzirom na to da su šeme povezivanja velikog formata, u okviru ovog rada biće prikazan samo deo aplikacije koji je dodeljen vertikalnom cilindru, dat na slici 5. Ulazi u blok tipa „VC_ECC“, pod nazivom „CYL_CTRL“, stižu od komunikacionih funkcijskih blokova, čiji je zadatak da čekaju i prenose događaje koji signaliziraju promene stanja drugih bitnih komponenata distribuiranog sistema. U ovom slučaju, prate se promene stanja horizontalnog cilindra.



Slika 5. Deo aplikacije koji je dodeljen vertikalnom cilindru.

Celokupna aplikacija distribuirana je na još 3 resursa (3 inteligentna aktuatora), koji ovde neće biti prikazani. Za kreirani sistem upravljanja u okviru FBDK okruženja, formirana je simulacija primenom funkcijskih blokova tipa OUT_BOOL. Nakon pokretanja simulacije, prikaz izgleda kao na slici 6 i po pritisku na taster „START“ simulacija započinje sa radom i moguće je pratiti promene stanja aktuatora, odnosno, rad simuliranog sistema. S obzirom na to da je u pitanju simulacija distribuiranog sistema, izabrano je da se svaki od aktuatora posmatra posebno, odnosno, da svaki od aktuatora ima sopstveni prozor.



Slika 6. Izgled pokrenute aplikacije koja simulira rad distribuiranog sistema upravljanja pneumatskim robotom.

Nakon uspešno izvedene simulacije, čime je potvrđena ispravnost razvijenog upravljanja, izvedeno je „prevođenje“ programa iz grafičkog jezika funkcijskih blokova u tekstualni jezik mikrokontrolera.

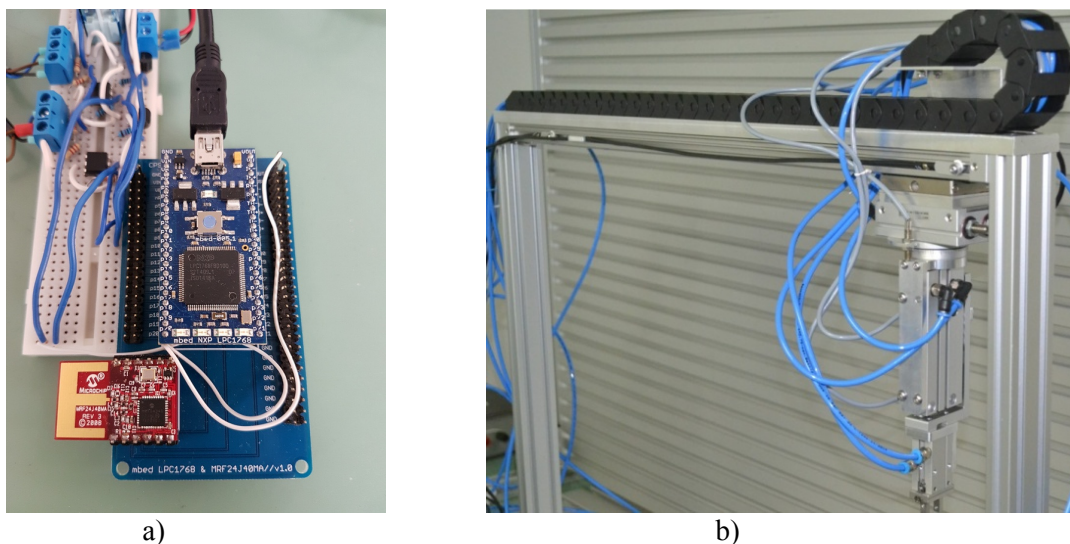
4. BEŽIČNI ČVOROVI

Trenutno, standard IEC 61499 još uvek je u fazi razvoja, predstavlja podršku projektovanju distribuiranog upravljanja i broj uređaja koji se mogu direktno programirati po pomenutom standardu je ograničen. Dostupna industrijska oprema najčešće je nekompatibilna i neophodno je izvršiti određene pripreme radnje koje omogućavaju primenu novog oblika upravljanja.

U okviru ovog rada, korišćena su 4 razvojna sistema mbed NXP LPC1768 [8], svaki sa 32-bitnim ARM Cortex M3 mikrokontrolerom, sa radnim taktom od 96 MHz. Pomenuti mikrokontroleri programiraju se u specijalnoj varijanti objektno-orijentisanog programskog jezika C++ [9] i izvorno ne podržavaju IEC 61499 standard, niti funkcijske blokove.

C++ izvorni kôd za mikrokontrolere napisan je po ugledu na aplikaciju razvijenu u okviru FBDK softvera, odnosno, određene C++ klase definisane su tako da emuliraju funkcijske blokove korišćene u okviru FBDK aplikacije kojom je inicijalno izvršena simulacija distribuiranog sistema upravljanja. S obzirom da su C++ i jezik funkcijskih blokova objektno orijentisani, postupak „prevođenja“ iz jezika funkcijskih blokova (standard IEC 61499) u C++ jezik mikrokontrolera (modifikovani standard C++17) bio je relativno jednostavan.

Za ostvarivanje bežične komunikacije između mikrokontrolera korišćeni su Microchip MRF24J40MA [10] integralni moduli koji podržavaju IEEE 802.15.4 komunikacioni standard, kolokvijalnog naziva ZigBee. Povezani par mbed LPC1768 i MRF24J40MA čini bežični čvor, sistem koji se može adresirati u bežičnoj mreži, koji može slati i primati podatke, a čije se ponašanje može programirati. Prikaz korišćenog bežičnog čvora dat je na slici 7.a.



Slika 7. a) Prikaz korišćenog bežičnog čvora. b) Pneumatski robot.

ZigBee komunikacioni standard predviđen je za komunikaciju između mašina i iz navedenog razloga je ograničen na male brzine prenosa podataka, do 256 kbps, što je izvedeno radi redukcije cene implementacije.

5. EKSPERIMENTALNA PROVERA

Tokom eksperimentalne provere svaki bežični čvor povezan je sa odgovarajućim aktuatorom kojim upravlja: 4 čvora – 4 aktuatora, čime su dobijena 4 inteligentna aktuatora. Eksperimentalni sistem prikazan je na slici 7.b. Iako su svi čvorovi hardverski međusobno isti, aktuatori su različiti i navedeni su u tabeli 1.

HC – horizontalni cilindar bez klipnjače, dvosmerni	SMC MZ3 B16-500
VC – vertikalni cilindar sa klipnjačom, dvosmerni	SMC MXH 10-60-Z
RC – rotirajući cilindar, dvosmerni	SMC MSQB7A
GR – hvatač (gripper), jednosmerni cilindar	SMC MHZ2-10DN

Tabela 1. Pregled korišćenih aktuatora.

Upravljački izvorni kôd za mikrokontrolere bežičnih čvorova napisan je u C++ jeziku i nakon kompilacije odgovarajući izvršni heksadecimalni kôd je sa računara kopiran u memoriju mikrokontrolera. Nakon

inicijalizacije, izvršene resetovanjem svih čvorova, sistem je pušten u rad, pritiskom na taster „START“. Sistem se ponašao kako je predviđeno, ispravno obavljajući saopšteni zadatak. Sva kretanja pneumatskog robota odvijala su se prema sekvenci (1), bez zastoja. Smatra se da je provera koncepta uspešno izvedena.

6. ZAKLJUČAK

U okviru ovog rada prikazane su neke od mogućnosti koje su otvorene novim IEC 61499 standardom, ilustrovane kroz projektovanje i primenu varijante distribuiranog sistema upravljanja baziranog na diskretnim događajima. Opisan sistem upravljanja predviđen je za upotrebu na pneumatskom robotu sa 3 stepena slobode (dva translatorna, jednim rotacionim) i hvatačem.

Simulacijom u okviru FBDK okruženja obavljeno je testiranje koncepta upravljanja. Prevođenjem jezika funkcijskih blokova u C++ jezik mikrokontrolera i proverom izvršnog programa na samim čvorovima (bez aktuatora) završeno je rano testiranje upravljačkog sistema. Povezivanjem čvorova sa odgovarajućim aktuatorima i testiranjem celokupnog mehatronskog sistema došlo se do povoljnih rezultata. Upravljački sistem, distribuiran na 4 inteligentna aktuatora, uspešno obavlja saopšteni zadatak kroz komunikaciju između čvorova, uz potpuno odsustvo centralizovanog upravljanja.

Bitno je napomenuti da narušavanje komunikacije između čvorova izazvano namernim ili nenamernim delovanjem može dovesti do nepravilnosti u radu sistema i ozbiljnih posledica po fizički deo sistema. Iako standard IEC 61499 može značajno doprineti industriji u dolazećem periodu, mora se voditi računa i o pratećoj infrastrukturi koja treba da omogući pouzdan i bezbedan rad. Dalje istraživanje trebalo bi da obuhvati unapređenje robusnosti sistema, fokusirajući se na periferiju oko IEC 61499 jezgra, s obzirom na to da distribuiran pristup upravljanju, zajedno sa primenom bežične komunikacije i interneta, otvara potpuno novu dimenziju bezbednosnih izazova koji su bili praktično nepostojeći u okviru „tradicionalnih“ izvedbi.

7. LITERATURA

- [1] Digital Transformation Monitor: *Germany: Industrie 4.0*, European Commission, January 2017.
https://ec.europa.eu/growth/tools-databases/dem/monitor/sites/default/files/DTM_Industrie%204.0.pdf
- [2] Zivana Jakovljevic, Stefan Mitrovic, Miroslav Pajic, *Cyber Physical Production Systems - An IEC 61499 Perspective*, Belgrade, Springer, 2017.
- [3] William MacDougall: *INDUSTRIE 4.0 Smart Manufacturing for the Future*, Berlin, Germany Trade and Invest, 2014.
- [4] Mitrović, S, Jakovljević, Ž: *Primena distribuiranog sistema upravljanja baziranog na IEC 61499 i 802.15.4 standardima*, ETIKUM 2017, 37-40, FTN, Novi Sad, Srbija, decembar 2017.
- [5] HOLOBLOC, INC – izvor FBDK softvera – <http://www.holobloc.com/>
- [6] Valeriy Vyatkin, *IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design*, 2nd Edition, ISA, 2012.
- [7] James H. Christensen, Thomas Strasser, Antonio Valentini, Valeriy Vyatkin, Alois Zoitl, *The IEC 61499 Function Block Standard: Overview of the Second Edition*, ISA, 2012.
- [8] mbed LPC1768 stranica <https://os.mbed.com/platforms/mbed-LPC1768/>
- [9] online mbed ARM kompajler <https://os.mbed.com/accounts/login/?next=%2Fcompiler%2F>
- [10] Microchip MRF24J40MA stranica <http://www.microchip.com/wwwproducts/en/MRF24J40MA>

Ispravnost svih linkova proverena 10.5.2018.

Mitrović, S, Jakovljević, Ž.

DISTRIBUTED CONTROL OF A PNEUMATIC ROBOT BASED ON IEC 61499 AND IEEE 802.15.4 STANDARDS

Abstract: *The market of today demands the development and application of new technologies that enable efficient small series production with extremely quick reaction time. Production must be flexible, with resources that have reconfigurable hardware and software structure. By decentralising the control and by distributing control tasks on multiple devices, configuration and reconfiguration of automated production systems is simplified, leading to significant savings in energy, material and time. This work presents distributed control system of a pneumatic painting robot based on IEC 61499 standard and IEEE 802.15.4 protocol application.*

Key words: *distributed control, IEC 61499, IEEE 802.15.4, ZigBee, pneumatic robot.*