

OPC UA na industrijskem krmilniku robota

Jan Perenič¹, Arko Erih², Marko Munih¹

¹Fakulteta za elektrotehniko, Univerza v Ljubljani, Tržaška c. 25, 1000 Ljubljana

²Yaskawa Slovenija, Lepovče 23, 1310 Ribnica

E-pošta: jan.perenic@gmail.com

Abstract

This work is focused on verifying the OPC UA protocol (Open Platform Communications Unified Architecture) applied to an industrial robot, that was developed by Yaskawa. This protocol provides a platform for digitization, inter-machine communication, and vertical connectivity to the cloud. Within this work, the open code program Node-RED was used as a client to aggregate the data in one place and display it in user interface. Except for very fast variable sampling, where Node-RED proved problematic, this approach achieved the anticipated functionality. Our user interface consists of three sections: basics, warnings, and joint. In addition to variables, we also have methods displayed in the application, as well as errors, jobs, and drive on/off. With this, all of the essential components for using OPC UA in an industrial setting are included.

1 Uvod

To delo uporablja trenutne komunikacijske protokole, ki se uporabljajo v industriji. Bolj podrobno se posveča protokolu OPC UA. Slednji je zaradi želje po poenoteni komunikaciji med različnimi platformami vedno bolj razširjen. Uporaba protokola je bila realizirana na primeru robotskega krmilnika proizvajalca Yaskawa. Podatki, prejeti preko protokola, so bili procesirani in prikazani na kontrolni plošči v sistemu Node-RED. Na ta način je bilo ponazorjeno, da je protokol zelo univerzalen, hiter in primeren za uporabo v industriji.

2 Protokol OPC UA

OPC Unified Architecture ali OPC UA je protokol, ki nudi veliko razširljivost, neodvisnost od platforme in varnost pri izmenjavi informacij v industrijskih sistemih. Je kompatibilen z vsemi računalniškimi operacijskimi sistemi in tudi z vgrajenimi mikrokrmilniki brez operacijskega sistema.

Cilj uporabe OPC UA, je večja interoperabilnost med napravami strojne opreme, programske opreme in zblíževanje različnih tehnologij. Uporablja se v industrijskih sistemih, kjer lahko skrbi in nadzoruje celotno proizvodnjo, nudi možnost oddaljene podpore, skrbi za kakovost in pridobivanje podatkov.

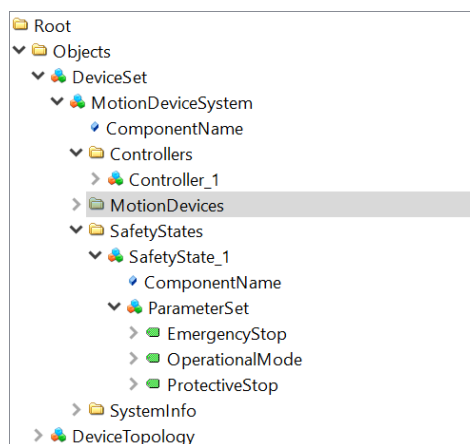
Razvil se je iz protokola OPC, ki je bil vezan na Windows. Ta ni imel vgrajene varne komunikacije in

dinamičnega pošiljanja podatkov med uporabniki. Imel je tudi hierarhično urejeno komunikacijo, kjer so vse informacije potekale skozi glavni računalnik. Tudi kodna osnova (ang. »code base«) je bila različna za aplikacije z različnimi specifikacijami.

Deluje na principu modulov, ki se kličejo za izvedbo določene funkcije. Nekateri so vgrajeni v sam protokol oz. jih definira proizvajalec strojne opreme. Vsak modul ima svojo funkcijo (npr. informacija o alarmih, opis naprave, ponastavitev alarmov ipd.). Moduli se lahko naredijo tudi po meri. To omogoča, da so prilagojeni glede potrebe uporabnika (ang. »companion models«).

Naprave, ki za komunikacijo uporabljajo protokol OPC UA, lahko podatke prenašajo prek internetnega omrežja. Informacije so transportirane z uporabo IP in SOAP protokola. To omogoča, da lahko naprave, ki nimajo nameščenega OPC UA protokola, vseeno sprejemajo podatke.

Vozlišče (ang. »node«) je osnovna enota podatkov v naslovnem prostoru OPC UA. Obstaja osem različnih razredov vozlišč. Najbolj pomembna so metode, spremenljivke in objekti. Z vozlišči protokol strežnikom zagotavlja standardni način za predstavljanje objektov odjemalcem. Vozlišča predstavljajo informacijo, sestavljeno iz atributov, dejanske vrednosti podatkov in ene ali več referenc na druga vozlišča. Vsako vozlišče ima edinstven ID. Ta je sestavljen iz URI (enolični identifikator vira), tipa podatka in identifikatorja. Vsako vozlišče pripada določenemu prostoru (ang. »namespace«). Tabela vseh prostorov je shranjena v sistemu, do katere lahko dostopajo vse naprave in prek tega kličejo prava vozlišča.



Slika 1. Hierarhična ureditev vozlišč v OPC UA

2.1 Prednosti:

- Decentralizacija: v starejših protokolih je hierarhija urejena piramidno, na vrhu naprava, ki prejema in pošilja vse podatke in nadzira celoten sistem. OPC UA pa je decentraliziran in omogoča prilagodljivo strukturo pri pridobivanju in prenosu podatkov.
- Neodvisnost protokola od operacijskega sistema.
- Scalability: omogoča upravljanje razširljivih sistemov brez dodatnih konfiguracij.
- Protokol samodejno v omrežju odkriva naprave in konfigurira nastavitve. Sistem je tako narejen na »plug-and-play« način.
- Interoperabilnost OPC UA omogoča končnim uporabnikom, da zgradijo industrijske sisteme po meri z uporabo strojne in programske opreme različnih proizvajalcev.

2.2 Primerjava z konkurenčnimi protokoli

OPC UA ne more v celoti nadomestiti protokola PROFINET. Ta je hitrejši in zagotavlja bolj zanesljiv prenos podatkov. PROFINET je hierarhično neprilagodljiv in zaprt v svoje omrežje. Lahko ga namestimo neposredno na PLC, kjer dobivamo direktne I/O podatke. OPC UA te možnosti nima, saj deluje na višjem sloju. Za pridobitev I/O podatkov potrebuje dodatno komunikacijo protokola tretjega proizvajalca. Oba pa lahko delujeta tudi skupaj. Tako eden skrbi za branje podatkov iz senzorjev in obdelavo, drugi pa za pošiljanje teh podatkov na strežnik. OPC UA protokol v celoti ne more nadomestiti niti EtherNet/IP, ki je namenjen predvsem prenosu podatkov iz I/O naprav na krmilnike. Protokol se še razvija, in ima v prihodnosti možnost izboljšati svoje slabosti ter tako nadomestiti večino svojih konkurenčnih protokolov.

Protokol OPC UA je narejen za prihodnost in industrijo 4.0, ki se nadalje razvija in izboljšuje. Ko bodo njegove slabosti odpravljene: konfiguracija različnih naprav in premajhna hitrost prenosa, lahko protokol nadomesti mnoge druge.

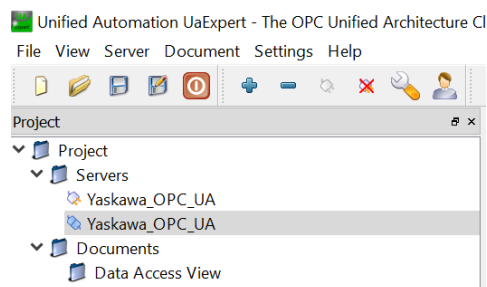
3 Strojna oprema

Za prototipno verzijo izvedbe je bil uporabljen šest osni robot MOTOMAN GP8. Ima nosilnost 8 kg in doseg 727 mm z ponovljivostjo +/- 0.01 mm. V tovarni je uparjen z robotskim krmilnikom. Do konca delovne dobe se ne ločita.

Potrebna je bila licenca za protokol OPC UA, ki jo je zagotovilo podjetje Yaskawa Slovenija. Naložena je bila na krmilnik robota. Računalnik se je povezal z robotskim krmilnikom prek Ethernet kabla. Računalniku je bil dodeljen stalen IP v lokalnem omrežju, prek katerega krmilnik in računalnik komunicirata.

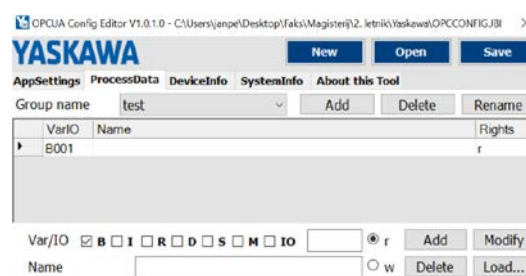
4 Programsko okolje

Program, ki vzpostavi povezavo na računalnik je Unified Automation UaExpert. Pri povezavi s krmilnikom se izpiše ureditev vseh vozlišč naprave, kot to ponazarja Slika 1. Navigacija po vozliščih je logično urejena. V primeru težav so podatki na voljo v dokumentaciji proizvajalca robotskega krmilnika. Tam so navedene poti do vseh vozlišč. Te podajajo želene vrednosti, ki jih kasneje prikazuje kontrolna plošča.



Slika 3. Prikaz vseh strežnikov in trenutno povezanega

Protokol OPC UA omogoča dodajanje lastnih spremenljivk. Dodajajo se prek internega Yaskawa programa OPCUA Config Editor, ki ga prikazuje Slika 4. Ta generira datoteko tipa .JBL, ki se naloži na krmilnik. Ko le ta prepozna dodane spremenljivke, jim lahko uporabnik dodeli zelene parametre robota, ki še niso vključeni v dokumentaciji. Ta funkcija je bila preverjena s štejetjem števila ponovitev, ko je robot v »Playback« načinu. »Playback« označuje stanje, ko robot ponavlja neko nalogo, brez dodatnih posegov operaterja. S spremenljivko »Playback« lahko štejemo število kosov, ki jih je robot obdelal.



Slika 4. Konfigurator za spremenljivke

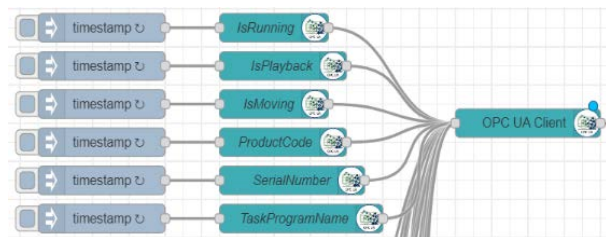
4.1 Node-RED

Za potrebe zbiranja vseh podatkov na enem mestu in načrtovanja vmesnika, prek katerega se nadzoruje stanje robota, se je uporabil Node-RED. Gre za odprtokodni (ang. »open-source«) program, ki temelji na Java Scriptu. Namenjen je povezovanju strojne opreme, API-jev in spletnih storitev. Zasnovan je predvsem za uporabo v Internet of Things, mogoča pa je uporaba tudi za druge namene, kot je OPC UA. Node-Red uporablja bločno programiranje. Z postavljanjem in povezovanjem blokov programiramo program. Blokom pravimo »nodes«. Program Node-RED deluje na računalnikih z

operacijskim sistemom Windows, Mac in Linux ter napravah brez operacijskega sistema. S povezovanjem blokov je dosežen »back-end« strani. Podatki se v nadaljevanju prikažejo prek uporabniškega vmesnika, kjer se lahko grafične komponente razporedijo po celotnem oknu brskalnika. Na ta način se ustvari želena oblika in funkcionalnosti.

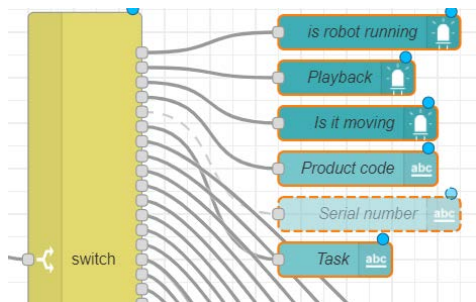
5 Rezultati

Prvi korak je bila vzpostavitev povezave Node-REDA z OPC UA. Za to je bil uporabljen blok OPC UA Client, ki je bil uporabljen za branje podatkov. Najprej se vanj vnese IP robotskega krmilnika, vrata prek katerih poteka komunikacija in še akcijo, ki jo robotski krmilnik izvaja. Na blok se poveže vse spremenljivke, ki se uporabijo v programu. Slika 5. prikazuje del celotnega programa. Ta se začne z blokom »timestamp«, ki pove kdaj mora računalnik prejeti informacijo iz robota in ali se ta akcija ponavlja ter na koliko časa. Čas vzorčenja je bil nastavljen na 1 s, saj je bilo pri manjših časih občasno mogoče zaznati preobremenitev trenutne izvedbe programa. Blok »timestamp« je povezan še na OPC UA Element blok, ki ima specifično določeni naslov spremenljivke, katera se bere. Informacije o specifični spremenljivki pošlje na element OPC UA Client.



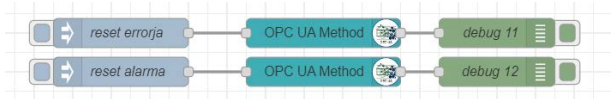
Slika 5. Povezava spremenljivk v blok za branje prek protokola OPC UA

Vse informacije, ki jih blok OPC UA Client (viden na sliki 5.) prejme, pošlje naprej v enem sporočilu. To sporočilo se loči na več različnih sporočil s komponento Switch, vidno na Sliki 6. Ta komponenta posamezna sporočila razpošlje na različne bloke, ki te informacije prikažejo na uporabniškem vmesniku. Spremenljivke se spreminjajo v realnem času tudi v grafičnem vmesniku.



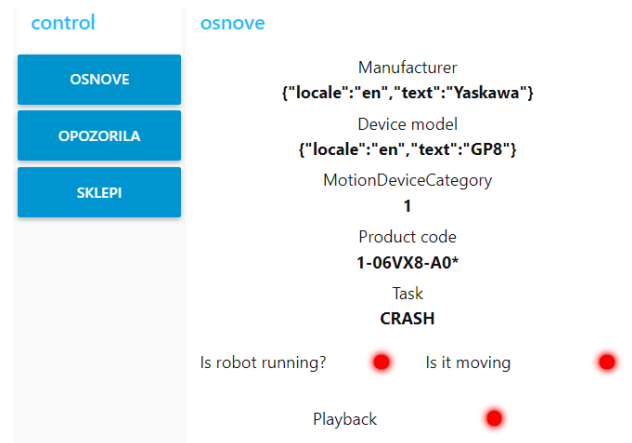
Slika 6. Nadaljevanje iz Slike 5., ki prikazuje razdelitev podatkov na komponente grafičnega vmesnika

Sam OPC UA sistem vsebuje tudi metode, ki omogočajo izvajanje določenih funkcij, kot so ponastavljanje napak (Error) in alarmov, nastavljanje naloge (poimenovanih »Job«) in vklop motorjev. V trenutnem delu je število teh metod omejeno, saj jih določa izvajalec. Z izvajanjem ene od metod se, s pritiskom na gumb na naši kontrolni plošči, metoda izvede na krmilniku. S to akcijo se bodo ugasnila Error sporočila (Error sporočilo se je prikazalo zaradi pritiska gumba »HOLD« na krmilniku, ko robot te akcije ni mogel izvesti, saj je bil v mirovanju) in pa ponastavili alarmi, če se je na primer robot ustavil zaradi detekcije trka. Ko se ponastavi alarm s težavo, se lahko opravljanje naloge nadaljuje.

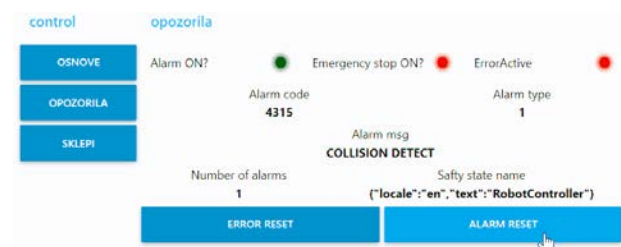


Slika 7. Klic metod v okolju Node-RED

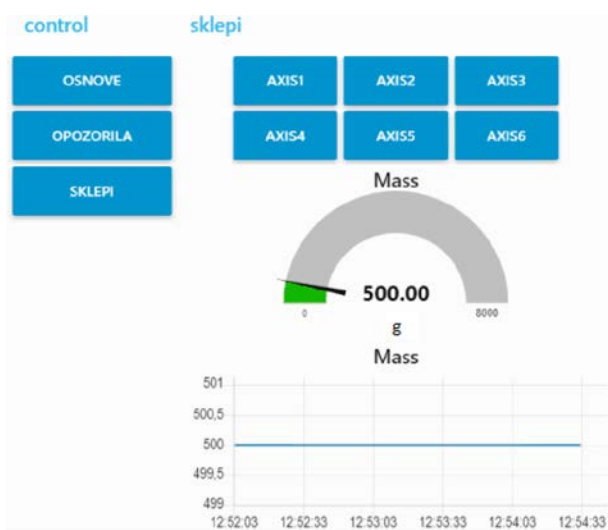
Sam vmesnik je preprost in pregleden. Zasnovan je na treh sklopih, in sicer »Osnove«, »Opozorila« in »Sklepi«. Vsak prikazuje določene informacije o svoji kategoriji. Med temi kategorijami izbirajo gumbi na levi strani. Informacije se prikazujejo s tekstom, lučkami, ki so prižgane, ko je pogoj »True«, in seveda grafi ter števc, ki nam grafično prikazujejo spreminjanje spremenljivke na nekem območju. V kategoriji Sklepi, prikazani na Sliki 10 in Sliki 11., je mogoča izbira med različnimi sklepi, za katere vmesnik prikaže specifične informacije.



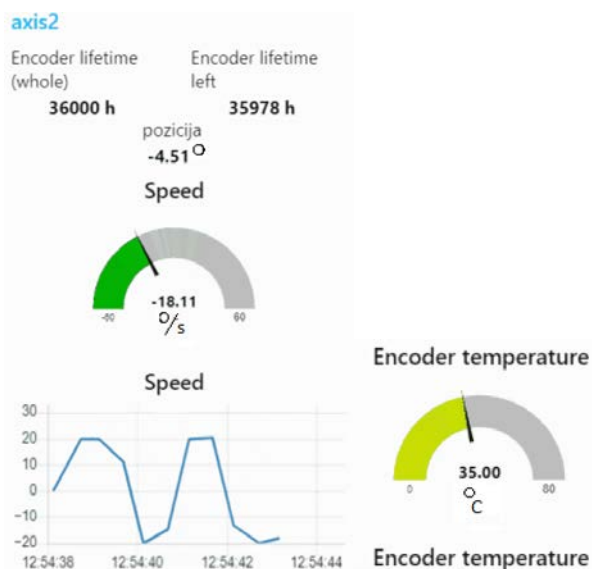
Slika 8. Kategorija Osnove uporabniškega vmesnika



Slika 9. Kategorija Opozorila uporabniškega vmesnika. Robot je zaznal trk



Slika 10. Kategorija Sklepi prikazuje informacijo o masi orodja na našem robotu



Slika 11. Kategorija Sklepi prikazuje informacije o sklepih

6 Diskusija

Protokol OPC UA je odličen protokol. Njegova lahka povezljivost in univerzalnost sta veliki prednosti, vendar pa ima OPC UA tudi svoje slabosti. V spletnih člankih veliko uporabnikov kot slabost navaja zahtevno konfiguracijo za vzpostavitev povezave z različnimi napravami različnih proizvajalcev na eno nadzorno ploščo. Tega trenutno nismo uspeli testirati, saj je bila na voljo le ena naprava in licenca le od proizvajalca Yaskawa, ki pa ni prenosljiva na druge naprave. Kljub tej pomanjkljivosti OPC UA omogoča hiter razvoj aplikacij, prek katere se spremljajo zelene informacije. Protokol je tudi varen in primeren za uporabnike začetnike. Povezava prek protokola in izdelava kontrolne plošče v programu Node-Red je preprosta, a hkrati omogoča veliko prostosti pri pridobivanju podatkov, njihovi obdelavi in prikazu.

Sam sistem pa je verjetno bolj primeren zgolj za testne namene. Ob zmanjšanju časa vzorčenja spremenljivk iz krmilnika, je Node-RED namreč zablokiral ali pa se je odklopil od komunikacije z protokolom. Izvedbe na več sistemih do sedaj nismo testirali. Predvideva pa se, da bi se v tem primeru pojavile težave pri zmožljivosti procesiranja podatkov.

Kot rešitev, ki jo navaja preučena literatura, bi se lahko uporabili programi, ki so namenjeni prikazu vmesnika prek OPC UA. Slabost teh za testne namene pa je seveda njihova cena. Še ena od možnih rešitev je uporaba programa LabView, ki ima možnost prejemanja podatkov prek protokola OPC UA. Ta je tudi bolj stabilen in primeren za industrijo.

Kontrolna plošča bi se lahko še dodatno nadgradila s pošiljanjem podatkov v sistem MES (Manufacturing execution system). Gre za računalniški sistem za nadzor in optimizacijo elementov proizvodnega procesa v realnem času. Optimizacija lahko poteka na več področjih od izdaje proizvodnega naloga do njegove realizacije. Namenjen je tudi povezovanju proizvodnega in poslovnega okolja. Sama implementacija ni zahtevna. V podjetju Yaskawa je bilo to preizkušeno pri štetju kosov, ki jih obdelava nek robot v obratu.

7 Zaključek

Protokol OPC UA je zelo uporaben pri načrtovanju nadzornih plošč, prek katerih lahko spremljamo podatke uporabniškega obrata. Protokol pa potrebuje še nekaj let, v katerih lahko njegovi razvijalci izpopolnijo delovanje. V prihodnosti bo tudi industrija začela pogosteje uporabljati protokol OPC UA, ki bo na ta način postal dostopnejši.

Viri

- [1] OPC Foundation [Online]. Dosegljivo: <https://opcfoundation.org/about/opc-technologies/opc-ua/>. [Dostopano: 4.12.2022].
- [2] M. Dobrić, A. Stare in S. Sokoliž, "Kaj prinaša OPC UA standard v praksi in kako pomaga pri digitalni transformaciji podjetja?", KONFERENCA Avtomatizacija v industriji in gospodarstvu (11 ; 2019 ; Maribor) [Online]. Dosegljivo: AIG 19 http://www.aig.si/19/zbornik/clanki/Sistemi_%20tehnologije_vodenja/Milan_Drobnic.pdf?fbclid=IwAR0lpEQQUjgXxl1BV3JQiMvHHVN39BHW3hl1Zl14EilYKN2rKW8_D2Zg-KQ [Dostopano: 16. 10. 2022].
- [3] P. Rebec. "Vmesniki OPC v industrijskih nadzornih sistemih," Diplomsko delo na visokošolskem strokovnem študiju. Ljubljana, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2011 [Dostopano: 16. 10. 2022].
- [4] Paessler The Monitoring Experts [Online]. Dosegljivo: https://www.paessler.com/opc-ua-monitoring?fbclid=IwAR1WmMKD8YAd6N41TY6jEkViASapUOkISWmvcOF1yi0JXGG4P_IL4BAptp0#opc-ua-definition [Dostopano: 29.11.2022]