

Praktično poučevanje načrtovanja in programiranja vgrajenih sistemov s pomočjo 6-nožnega robota.

Jernej Olenšek

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: jernej.olensek@uni-lj.si

Povzetek. V prispevku je predstavljen razvoj 6-nožnega robota (Hexapod), ki bo služil kot prilagodljiva platforma za praktično delo za študente elektrotehnike. Cilj projekta je študentom ponuditi realen relativno kompleksen sistem, na katerem bodo lahko spoznali različne vidike integracije strojne in programske opreme pri razvoju vgrajenih sistemov. Zasnova robota ponuja veliko možnosti za nadgradnjo, pri čemer je bilo precej pozornosti namenjene enostavnosti vzdrževanja in zaščiti pred napakami pri programiranju in/ali vodenju robota. Tem se namreč pri učenju običajno ne moremo izogniti in so hkrati tudi najboljši vir izkušenj za bodoče inženirje.

Ključne besede: vgrajeni sistemi, 6-nožni robot, hexapod, programiranje

A hexapod robot for an embedded system design and programming course

The paper presents a design of a six-legged robot (hexapod) that will serve as a flexible platform for practical exercises for electrical engineering students. The aim is to offer students a realistic fairly complex system, enabling to learn about various aspects of a hardware and software integration in the process of the embedded system design. The robot design offers many possibilities for upgrades, with a considerable attention paid to the ease of maintenance and protection against errors in programming and/or operating the robot. These are often unavoidable during the learning process and are at the same time the best source of experience for future engineers.

Keywords: embedded systems, 6-legged robot, hexapod, programming

1 UVOD

Načrtovanje in programiranje vgrajenih sistemov je lahko precej kompleksna naloga [1]. Od inženirja zahteva širok spekter znanja, ki mora pogosto pokrivati več področij kot so elektronika, krmilni in napajalni sistemi, komunikacijski protokoli, obdelava signalov, krmiljenje senzorjev in aktuatorjev ter seveda programiranje.

Poučevanje na tem področju je postalo izjemno pomembno za bodoče inženirje, saj se vgrajeni sistemi v vsakdanjem življenju pojavljajo tako rekoč na vsakem koraku. Težava, s katero se pogosto soočamo pri poučevanju te tematike, so razlike v praktičnih izkušnjah in ravni predznanja študentov na področju elektrotehnike in programiranja. To je še zlasti opazno pri izbirnih predmetih, kjer so v isti skupini študentje različnih študijskih smeri. Na Fakulteti za elektrotehniko v Ljubljani se vsi študenti že v prvem letniku srečajo z osnovami

programiranja, kjer poskušajo razviti algoritemski način razmišljanja in reševanja problemov. V višjih letnikih se smeri razdelijo po tematskih področjih, ki niso vsa enako pomembna pri razvoju vgrajenih sistemov.

Zato je za študente ključno, da jim program predmetov poleg predavanj omogoča tudi čim več praktičnega dela [2], [3]. Pri predmetih načrtovanja in programiranja vgrajenih sistemov morajo študenti pred ustnim izpitom uspešno zagovarjati laboratorijske vaje, vendar samo reprodukcija snovi z vodenih vaj ne prinaša dragocenih izkušenj in znanj, ki jih lahko pridobijo le s praktičnim samostojnim delom na realnem projektu. Zato imajo študenti možnost izpeljave samostojnega projekta, za katerega morajo pripraviti specifikacije, ga izdelati, pripraviti poročilo oz. dokumentacijo, ter projekt tudi predstaviti.

Da bi čim več študentov spodbudili k izdelavi samostojnih projektov, moramo že laboratorijske vaje izvajati na kar se da zanimiv in praktično uporaben način z vnaprej zastavljenim realnim ciljem.

Zato smo za laboratorijske vaje razvili mobilen in fleksibilen sistem, ki bo služil kot platforma za praktično delo in tudi samostojne študentske projekte. Odločili smo se za 6-nožnega robota (Hexapod), s katerim želimo študentom ponuditi možnost, da se neposredno spopadejo z realnim izzivom integracije raznovrstne strojne in programske opreme pri realnem projektu, pri tem pa jim prihranimo strošek nabave materiala in elektronskih komponent. S takim pristopom je namreč precej lažje vzbuditi in tudi ohraniti zanimanje za aktivno sodelovanje pri izobraževalnem procesu in pozneje pri njihovih samostojnih projektih.

Prispevek je organiziran na sledeči način. Uvodnemu poglavju sledi opis osnovnih specifikacij robota, ki so bile glavno vodilo pri izbiri sistema za praktične

vaje. Poglavlje 3 opisuje modularno zasnovano robota, v poglavju 4 pa je na kratko predstavljeno osnovno matematično ozadje, na katerem temeljijo izračuni za krmiljenje robota. Poglavlje 5 vsebuje nekaj podrobnosti o zaščiti in kalibraciji robota, v poglavju 6 pa se dotaknemo tudi večopravnosti, ki je pomemben del programiranja vgrajenih sistemov. Poglavlje 7 predstavlja različne načine upravljanja robota, v poglavju 8 pa je predstavljen nekaj idej za nadgradnje oziroma samostojne študentske projekte. Na koncu prispevka sledi še povzetek in nekaj zaključnih misli.

2 IZBIRA SISTEMA

V praksi so majhni mobilni roboti zelo uporabni pri raziskovanju težko dostopnih krajev in krajev, ki za človeka niso primerni [4]. Uporabljajo se lahko pri iskanju ponesrečencev ob naravnih nesrečah ali pri pregledu in popravilih večjih naprav, motorjev, itd., saj se lahko gibajo po zelo razgibanem terenu, in so lahko dovolj majhni, da zlezejo tudi skozi zelo majhne odprtine.

Več-nožni roboti pri gibanju posnemajo gibanje živali in so v primerjavi z roboti na kolesih običajno počasnejši. Poleg tega imajo kompleksnejšo konstrukcijo in višjo ceno ter potrebujejo tudi zahtevnejše krmiljenje. Vendar pa imajo roboti s kolesi običajno težave pri gibanju po neravnem in nepravilnem terenu. Neodvisno krmiljenje nog več-nožnih robotov in večje število prostostnih stopenj (število motorjev) je pri težavnih terenih ključno za zagotavljanje gibljivosti in stabilnosti pri gibanju.

Seveda tudi več-nožni roboti obstajajo v različnih konfiguracijah, ki imajo vsaka svoje prednosti in slabosti. Za 6-nožnega robota je na primer lažje zagotoviti stabilnost kot za 2-, 3-, ali 4-nožnega robota, je pa zahtevnejši za izdelavo in vzdrževanje ter seveda tudi dražji.

Glavni kriterij pri izbiri sistema za praktične vaje je bila njegova kompleksnost. Ponujati mora možnosti za praktično delo vsem študentom, ne glede na njihovo predznanje s področja elektrotehnike in/ali programiranja. Predvsem mora omogočati osnovno spoznavanje z vgrajenimi sistemi in posameznimi perifernimi enotami. Analogno digitalno pretvorbo na primer lahko spoznamo na preprostem generičnem primeru s potencijetrom in izpisom na zaslon ali pa na nekoliko bolj zanimiv in praktično uporaben način, kot je spremljanje stanja baterije ali odčitavanje lege igralnih palic (angl. joystick) za krmiljenje robota. Poleg osnov pa mora biti sistem dovolj velik izziv tudi za izkušenejše študente in spoznavanje nekoliko bolj zahtevnih konceptov, kot so večopravnost, časovna determinističnost itd. Pri izbiri konfiguracije in komponent je treba upoštevati še nekaj dodatnih kriterijev:

- robot mora biti dovolj velik in zmogljiv za razširitve oz. vgradnjo dodatnih komponent in elektronike (npr. kamera, senzorji ...),

- zasnova mora biti dovolj robustna, da se zmanjšajo potrebe po menjavi delov zaradi obrabe ali poškodb,
- nujna je določena stopnja zaščite pred napakami med programiranjem ali vodenjem,
- za daljše obratovanje med vajami je potrebno dovolj zmogljivo baterijsko napajanje,
- cena.

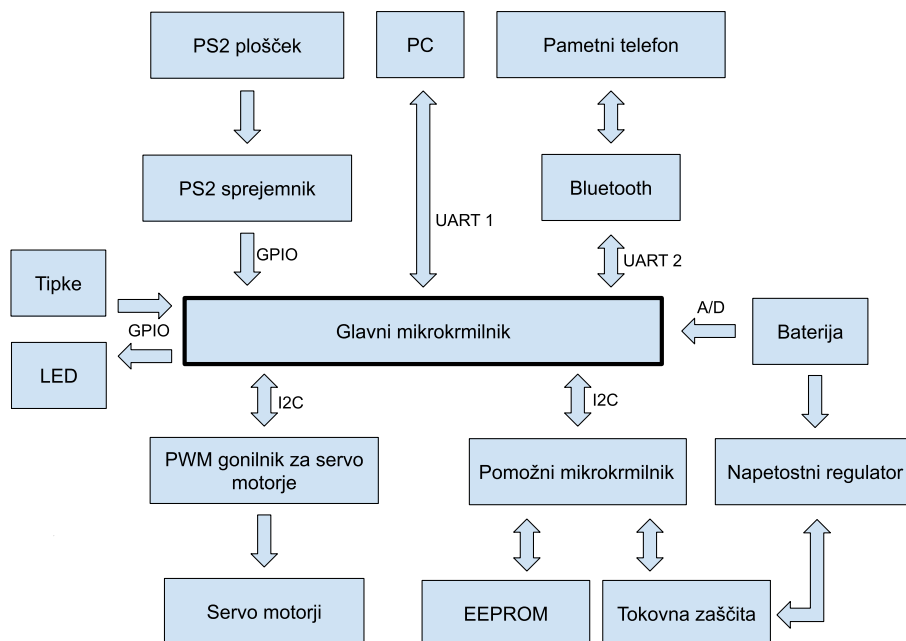
Izbran 6-nožni robot vsebuje dovolj komponent, da omogoča spoznavanje množice konceptov, ki se pogosto uporabljajo pri načrtovanju vgrajenih sistemov, na primer:

- splošni koncepti mikrokrmilnikov in vgrajenih sistemov (registri, vodila, pomnilnik, periferne enote ...),
- programiranje v jeziku C na osnovnem in tudi višjem nivoju,
- vzporedna digitalna vrata (GPIO),
- analogno-digitalni pretvornik (ADC),
- asinhrona serijska komunikacija (UART),
- sinhrono komunikacijsko vodilo I²C (angl. Inter Integrated Circuit, tudi I2C),
- časovniki in časovna determinističnost,
- prekinitve,
- večopravnost, vzporedno izvajanje opravil, operacijski sistem v realnem času (na primer FreeRTOS [9]),
- krmiljenje motorjev, pulzno-širinska modulacija (PWM),
- brezžična komunikacija z napravo Bluetooth,
- napajanje in zaščita komponent pred preobremenitvijo itd.

Izbira konkretnega mikrokrmilnika ni bil zelo pomemben kriterij, saj je pri praktičnih vajah glavni poudarek na konceptih, ki so skupni vsem vgrajenim sistemom. Poleg tega se področje mikrokrmilnikov precej hitro razvija, zato je dobrodošla možnost, da po potrebi mikrokrmilnik kadarkoli zamenjamo z novejšim. Trenutno je v sistem vgrajen mikrokrmilnik Microchip SAM3X8E z jedrom ARM Cortex-M3, vendar zasnova robota omogoča enostavno in hitro adaptacijo z drugim mikrokrmilnikom.

3 ZASNOVA ROBOTA

Robot je namenjen praktičnemu delu v sklopu laboratorijskih vaj, zato je upravičena predpostavka, da bo pogosto prihajalo do napak pri programiranju in/ali vodenju robota. S tem so povezane tudi precejšnja fizična obraba in celo poškodbe posameznih sestavnih delov. Zato je bila ena od prioritet pri načrtovanju robota modularna zasnova, ki bo brez večjih fizičnih posegov v celoten sistem omogočala enostaven dostop do posameznih neodvisnih modulov za namene vzdrževanja, popravljanja in nadgrajevanja sistema. Pri tem smo nekoliko žrtvovali



Slika 1: Blokovna shema robota.

kompaktnost sistema v zameno za dostopnost posameznih komponent.

Robot je sestavljen iz treh ločenih, neodvisnih modulov, ki si delijo le povezave za napajanje in komunikacijsko vodilo I2C. Napajalni modul vsebuje dva stikalna napajalnika in vso elektroniko, ki omogoča tokovno zaščito pred preobremenitvijo. Nadzor te mora biti neodvisen od glavnega programa robota in je izveden s pomočjo pomožnega mikrokrmilnika ATmega328p, ki s svojim vgrajenim pomnilnikom EEPROM služi tudi kot trajni pomnilnik za shranjevanje kalibracijskih parametrov posameznega izdelanega robota. Tudi te želimo namreč po kalibraciji ločiti od morebitnih napak pri programiranju glavnega programa robota.

Drugi modul je izdelan na ločeni ploščici PCB (ang. Printed Circuit Board), na kateri je združena vsa elektronika skupaj z dvema gonilnikoma PWM (pulzno-širinska modulacija, angl. pulse width modulation), ki skrbita za krmiljenje 18 servo motorjev.

Glavni mikrokrmilniški modul je prav tako izveden na ločeni ploščici PCB in vsebuje mikrokrmilnik, modul Bluetooth ter nekaj digitalnih tipk in signalnih diod LED. Komunikacija s preostalima dvema moduloma poteka preko sinhronnega vodila I2C, kar pomeni poleg napajanja le še dve dodatni povezavi (SDA – podatki in SCL – ura) za nadzor vseh komponent sistema.

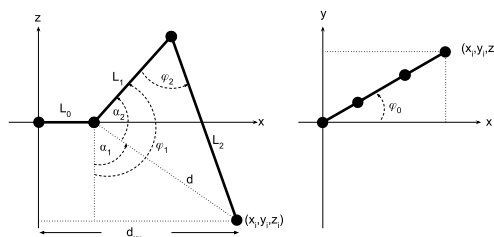
Blokovna shema robota je prikazana na sliki 1. Za izdelavo robota smo uporabili naslednje komponente:

- aluminijasto ogrodje,
- mikrokrmilnik AT91SAM3X8E,
- pomožni mikrokrmilnik ATmega328p,
- Bluetooth modul HC-05 z master/slave načinom,
- dva 16 kanalna PWM gonilnika PCA9685,

- dva nastavljiva stikalna napajalnika XL4005,
- dva tokovna senzorja z ojačevalnikom TSC101C,
- 18 servo motorjev (MG996R),
- elektronske komponente (upori, kondenzatorji, tipke, LED-diode, ...),
- akumulator LiPo.

4 MATEMATIČNO OZADJE

Robot je simetričen glede na x in y os ter ima skupaj 18 prostostnih stopenj – vsaka od šestih nog ima tri sklepe, ki jih nadzirajo servo motorji (slika 2). Krmiljenje položajev servo motorjev oziroma kotov sklepov $\varphi_0, \varphi_1, \varphi_2$ zahteva tri vhodne podatke za vsako nogo. Za naš konkreten primer bo sistem enolično analitično rešljiv, če za konico vsake noge poznamo vse tri koordinate (x_i, y_i, z_i) v prostoru. Postopek izračuna kotov sklepov je znan pod imenom inverzna kinematika [5] in ga lahko z nekaj znanja geometrije in trigonometrije zapišemo v naslednji obliki:



Slika 2: Inverzna kinematika: izračun kotov motorjev iz znane pozicije nog.

$$\begin{aligned}\varphi_0 &= \arctan\left(\frac{x_i}{y_i}\right) \\ d_{xy}^2 &= x_i^2 + y_i^2 \\ d^2 &= z_i^2 + (d_{xy} - L_0)^2 \\ \alpha_1 &= \arctan\left(\frac{d_{xy} - L_0}{z_i}\right) \\ L_2^2 &= L_1^2 + d^2 - 2 \cdot L_1 \cdot d \cdot \cos(\alpha_2) \\ \varphi_1 &= \alpha_1 + \alpha_2 \\ d^2 &= L_1^2 + L_2^2 - 2 \cdot L_1 \cdot L_2 \cdot \cos(\varphi_2)\end{aligned}$$

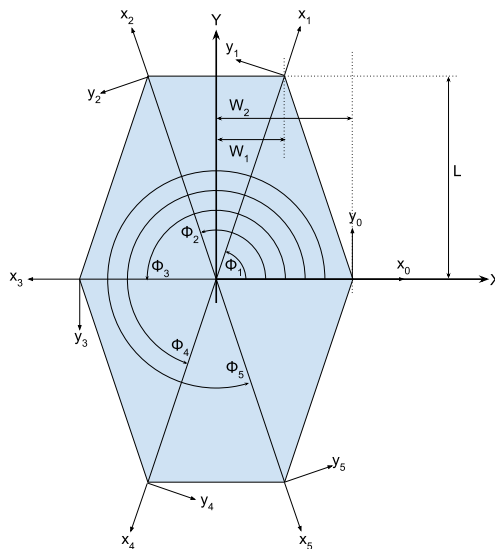
kjer L_0, L_1, L_2 predstavljajo dolžine 3 segmentov noge: kolk (angl. coxa), stegno (angl. femur), golen (angl. tibia). Iz zgornjih enačb lahko enolično določimo kote sklepov $\varphi_0, \varphi_1, \varphi_2$ s katerimi so nato določena stanja servo motorjev. Pri izračunu je kot izhodišče za koordinate i -te noge (x_i, y_i, z_i) upoštevana točka pritrditve noge, x-os pa je usmerjena v smeri od težišča robota do točke pritrditve (slika 3).

Za usklajeno gibanje nog je treba njihov položaj določati glede na koordinatni sistem, ki ga določa težišče robota. Pri izračunu položaja motorjev moramo torej upoštevati še linearen premik točke pritrditve noge (W_1 oz. W_2 ter L) in zasuk koordinat noge okoli vertikalne z osi (ϕ_i), glede na globalni koordinatni sistem robota (slika 3). Če k temu dodamo še premik težišča robota in obračanje okoli vseh treh osi (usmerjen pogled), gre za relativno kompleksen izračun, ki ga je treba izvajati za vsako nogo posebej pri vsakem premiku robota. Pri tem je kritična komponenta tudi časovna usklajenost vseh gibov, saj lahko le tako zagotovimo gladko in zvezno gibanje celega robota.

Koordinacija premikanja vseh šestih nog robota pomeni svojevrsten problem, saj tak robot omogoča več različnih načinov premikanja [6]. Za stabilno gibanje je treba v vsakem trenutku zagotavljati ustrezno lego težišča robota, ki mora ves čas ležati nad poligonom podpore, ki ga tvorijo noge, ki so v stiku s podlago. Običajno gibanje na tak ali drugačen način posnema gibanje žuželk na primer:

- najbolj stabilna vendar tudi najpočasnejša je hoja s podporo petih nog, ki spominja na valovanje (t. i. wave gait) in je najprimernejša za neraven in razgiban teren;
- najhitrejši še statično stabilen način premikanja je trinožna hoja (t. i. tripod gait), pri kateri je v vsakem trenutku robot podprt s tremi nogami. Ta način je primeren predvsem za raven teren;
- hoja s podporo štirih nog (t. i. ripple gait) je po hitrosti in primernosti za neraven teren nekje med zgornjima dvema.

Trenutno je implementiran način trinožne hoje po ravnem terenu, pri čemer lahko upravljavec spreminja višino telesa, hitrost hoje in dolžino korakov ter intenzivnost oziroma višino korakov pri dvigu nog. Upravljavec lahko neodvisno določi poljubno smer hoje v x-y



Slika 3: Parametri robota

ravnini in jo poljubno kombinira z obračanjem oziroma kroženjem robota okoli vertikalne z-osi. Poleg osnovne hoje robot omogoča tudi linearen premik težišča telesa in spremembo orientacije v prostoru oziroma rotacijo telesa, kar omogoča neodvisno usmerjanje pogleda med hojo. Ne nazadnje pa lahko upravljavec neodvisno upravlja tudi vsako od 6 nog.

Implementacija drugih načinov premikanja je prepuščena študentom za vajo oziroma kot samostojni projekt. Podobno velja za gibanje po neravnem terenu, saj zahteva uporabo dodatnih senzorjev in posebne načine načrtovanja gibov (angl. motion planning, path planning) [6], [7]. Zagotavljanje zveznega gladkega gibanja in stabilnosti robota je v teh primerih bolj zahteven problem.

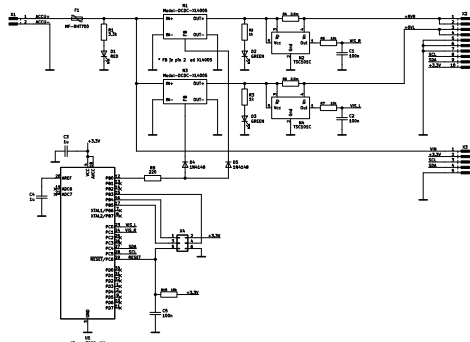
5 ZAŠČITA IN KALIBRACIJA ROBOTA

Posebna pozornost je bila posvečena napajanju in zaščiti robota pred preobremenitvijo zaradi morebitnih napak pri programiranju in/ali vodenju robota. Dva stikalna napajalnika XL4005 skrbita za prilagoditev baterijske napetosti na ustrezne napetostne nivoje. Tokovna zaščita je izvedena s pomočjo dveh senzorjev toka TSC101C, nekaj dodatne elektronike in neodvisnega mikrokontrolerja, ki omogoča programski izklop napajanja in nenehno spremlja jakost električnega toka obeh stikalnih napajalnikov (slika 4). Ta sicer že vsebujeta integrirani sistem za samodejni izklop, vendar je bil zaradi podaljšanja življenjske dobe komponent naš cilj izklop izvesti, še preden pride do pregrevanja. V primeru daljših večjih obremenitev se tako s pomočjo pomožnega mikrokontrolerja ATMega328p napajanje sistema za določen čas programsko izklopi.

Vse parametre tokovne zaščite (največji dovoljen tok, način vzorčenja tokovne obremenitve, čas trajanja izklopa itd.) pomožni mikrokontroler hrani v svojem vgra-

jenem pomnilniku EEPROM. Dvosmerna komunikacija z glavnim programom robota poteka preko vodila I2C. Implementirani so programski vklop/izklop napajanja, spremljanje tokovne obremenitve v realnem času in spreminjanje parametrov tokovne zaščite. Večino časa pa pomožni mikrokrmilnik deluje neodvisno, saj tokovna zaščita ne sme biti odvisna od morebitnih napak pri programiranju glavnega programa robota.

EEPROM pomožnega mikrokrmilnika je izkoriščen tudi za trajno shranjevanje specifičnih parametrov posameznega izdelanega robota. Vsakemu robotu je dodeljena unikatna oznaka (ID), ki med drugim služi za identifikacijo v brezžičnem omrežju Bluetooth. Poleg tega je izdelava večjega števila popolnoma identičnih robotov v praksi težko izvedljiva. Zato je za vsakega robota potrebna kalibracija oz. umerjanje. Pri tem postopku za posameznega robota določimo nekatere specifične parametre (npr. kalibracija zamikov kotov motorjev zaradi fizičnih razlik pri pritrjevanju motorjev), ki jih je treba zapisati v trajni pomnilnik. EEPROM pomožnega mikrokrmilnika se ponuja kot priročna rešitev za to nalogo, saj tako študenti med programiranjem glavnega programa precej težje po pomoti (ali namerno) spremenijo te parametre, kar bi vodilo v zamudno ponovno kalibracijo.



Slika 4: Napajalni modul s tokovno zaščito.

6 RTOS

Za nekoliko bolj kompleksne sisteme postane večopravilni operacijski sistem v realnem času (RTOS, angl. real time operating system) skoraj nepogrešljiv, saj je sicer težko zagotavljati sočasno izvajanje več nalog in hkrati skrbeti še za determinističnost časovno kritičnih opravil.

V našem primeru je treba sinhrono in v natančno določenih trenutkih izračunavati položaje nog in krmiliti 18 motorjev (postopek inverzne kinematike), sicer gibanje robota ne bo zvezno in gladko. Hkrati se mora sistem dovolj hitro odzivati na ukaze upravljalca, ki prihajajo bodisi preko povezave Bluetooth, igralnega ploščka PS2 ali serijske povezave UART z osebnim računalnikom. Poleg tega je treba še krmiliti indikatorske

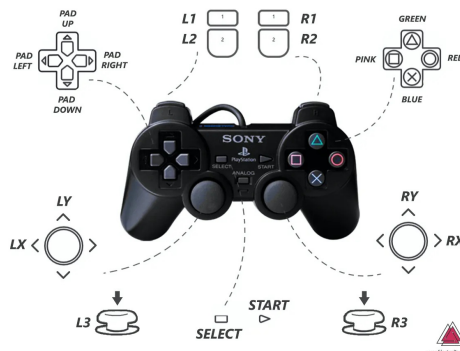
LED-diode in zaznavati pritiske tipk ter se ustrezno odzvati. Dodatno lahko poskrbimo za nadzorovan izklop sistema ob preveliki obremenitvi še preden se sproži sistem samodejne tokovne zaščite. Ne nazadnje pa je treba paziti tudi na stanje baterije, saj je preprečevanje prevelike izpraznjenosti baterije nujno za njeno dolgo življensko dobo.

Opravil je torej kar precej, nekatera pa so za gladko gibanje robota tudi časovno kritična. Gre torej za večopravilni sistem, v katerem lahko študenti praktično spoznajo tudi lastnosti realno-časnih operacijskih sistemov (na primer FreeRTOS [9]).

7 KRMILJENJE ROBOTA

Robota lahko krmilimo na različne načine. Serijska povezava z osebnim računalnikom je nujna v fazi razvoja, za kalibracijo, spreminjanje parametrov sistema in razhroščevanje. Pri normalni uporabi pa je bolj smiselna vpeljava brezžičnega nadzora. Na voljo je nadzor s pomočjo pametnega telefona (ali osebnega računalnika) preko povezave Bluetooth. Če naprava Bluetooth ni na voljo, lahko za krmiljenje uporabimo tudi priljubljen igralni plošček PS2 (slika 5). Ti obstajajo v žični in brezžični različici, za obe pa je možno nabaviti sprejemnik, ki ga je relativno enostavno priključiti na mikrokrmilnik. Za komunikacijo z mikrokrmilnikom pa je seveda potrebno napisati ustrezen gonilnik (angl. driver).

Za izobraževalne namene pa je najboljša izbira izdelava lastnega daljinskega upravljalnika Bluetooth po vzoru ploščka PS2. Potrebujemo mikrokrmilnik, dve igralni palici (angl. joystick), oddajnik/sprejemnik Bluetooth in nekaj tipk (plošček PS2 ima 16 tipk). Za bolj celovit vpogled v delovanje robota lahko dodamo še zaslon LCD za izpis telemetrije in raznih sporočil robota. Tako imajo študenti na vajah možnost načrtovati in izdelati povsem realen vgrajen sistem, kar je precej bolj zanimivo kot suhoparno spoznavanje posameznih perifernih enot mikrokrmilnika.



Slika 5: igralni plošček PS2

Vir: <https://www.hackster.io/electropeak/how-to-interface-ps2-wireless-controller-w-arduino-a0a813>



Slika 6: Izdelan robot.

8 NADGRADNJE

Izdelan robot (slika 6) ponuja študentom veliko možnosti tudi za nadgradnje, saj je zasnova dovolj robustna in omogoča enostavno dodajanje elektronike in tudi večjih modulov. Poleg že omenjene izdelave daljinca se lahko lotijo tudi zahtevnejših projektov, kot je dodajanje senzorjev za zaznavanje dotika nog s podlago. Temu lahko sledi sistem za samodejno gibanje po neravnem terenu ali stopnicah, ki ga lahko dodatno nadgradijo s senzorjem nagiba in tako zagotovijo vodoraven položaj telesa robota ne glede na konfiguracijo terena oz. podlage. Zaznavanje okolice s pomočjo kamere, lidar-ja ali ultrazvoka lahko vodi v razvoj sistema za samodejno izogibanje oviram, ki ga lahko pozneje dopolnijo še z navigacijskim sistemom (npr. GPS). Podobnih možnosti nadgradnje je torej več kot dovolj tudi za bolj izkušene in zagrete študente.

9 ZAKLJUČEK

V prispevku je predstavljen 6-nožni robot (Hexapod), ki smo ga razvili za namene poučevanja načrtovanja in programiranja vgrajenih sistemov. Z njim študentom ponujamo praktično delo pri realnem projektu, hkrati pa jim prihranimo strošek nabave materiala. Tak način učenja je učinkovitejši kot klasične laboratorijske vaje z včasih suhoparnim spoznavanjem posameznih perifernih enot mikrokrmilnika. Sistem je izdelan do točke, ki omogoča delo vsem študentom, ne glede na njihov nivo predznanja elektrotehnike in programiranja. Še vedno pa je dovolj kompleksen, da predstavlja izziv tudi izkušenejšim študentom, ki se lahko lotijo nadgradnje z dodatno elektroniko in implementacijo novih funkcionalnosti. Izvedbo praktičnih vaj precej olajša modularna zasnova robota, ki omogoča enostavno in hitro vzdrževanje, popravljanje in nadgrajevanje posameznih komponent z minimalnim poseganjem v druge, že dokončane module.

ZAHVALA

Raziskavo je sofinancirala Javna agencija za raziskovalno dejavnost Republike Slovenije v okviru programa P2-0246 - Algoritmi in optimizacijski postopki v telekomunikacijah.

Posebno zahvalo si zasluži Brane Ždralo, ki je s svojim znanjem in izkušnjami pri razvoju in izdelavi elektronskih vezij močno olajšal izvedbo projekta.

10 REFERENCE

LITERATURA

- [1] Peter Bertels. Et. al., *Teaching skills and concepts for embedded systems design*, *Journal of SIGBED Review*, Vol. 1, pp. 1–4, 2009.
- [2] A. F. Mondragon-Torres, J. W. Christman, *A comprehensive embedded systems design course and laboratory*, *2013 IEEE International Conference on Microelectronic Systems Education (MSE)*, pp. 56–59, 2013, doi: 10.1109/MSE.2013.6566704.
- [3] R. Manseur, *Studio teaching: An embedded systems course*, *2018 IEEE Global Engineering Education Conference (EDUCON)*, pp. 590–598, 2018, doi: 10.1109/EDUCON.2018.8363284.
- [4] K. Asamura, S. Nagasawa, *A micro hexapod robot for swarm applications assembled from a single FPC sheet*, *JAPANESE JOURNAL OF APPLIED PHYSICS*, Vol. 60, No. SC, SCCL03, 06.2021. doi: 10.35848/1347-4065/abe689.
- [5] K. Priandana, A. Buono, Wulandari, *Hexapod Leg Coordination using Simple Geometrical Tripod-Gait and Inverse Kinematics Approach*, *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 35–40, 2017, doi: 10.1109/ICACSIS.2017.8355009.
- [6] Fusheng Zha, Chen Chen, Wei Guo, Penglong Zheng, Junyi Shi, *A free gait controller designed for a heavy load hexapod robot*, *Advances in Mechanical Engineering*, Vol. 11(3) pp. 1–17, 2019, doi: 10.1177/1687814019838369
- [7] Q. Liu, T. Jing, *A Survey on Hexapod Walking Robot and Gait Planning*, *Proceedings of the 2015 International Forum on Energy, Environment Science and Materials*, pp. 356–364, 2015, doi: 10.2991/ifeesm-15.2015.67
- [8] P. Čížek, J. Kubík, J. Faigl, *Online Foot-Strike Detection Using Inertial Measurements for Multi-Legged Walking Robots*, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7622–7627, 2018, doi: 10.1109/IROS.2018.8594010.
- [9] *FREERTOS*, <https://www.freertos.org/>.
- [10] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: modelling, planning and control*, Springer, 2009,

Jernej Olenšek je leta 2009 doktoriral s področja elektrotehnike na Univerzi v Ljubljani, na Fakulteti za elektrotehniko je zdaj zaposlen kot asistent. Njegovo raziskovalno področje zajema numerično simulacijo, teorijo in uporabo optimizacijskih metod, ter uporabo računalniških metod pri načrtovanju elektronskih vezij.