KATALOG ZNANJA

1. Ime modula: RAČUNALNIŠKO VODNJE PROCESOV (RVP4)

2. Usmerjevalni cilji:

Dijak:

- se spozna z načinom programiranja za čim bolj optimalno porabo procesorske moči (ne uporablja zakasnilnih zank s ponavljanjem enostavnih strojnih ukazov, ampak organizira programske rešitve z upoštevanjem različnih sprememb in dogodkov v procesu),
- povezuje hardversko okolje s programskim upravljanjem preko krmilnika,
- odkriva napake v povezovalnih elementih, namestitvenih ploščah, elektronskih komponentah, programski kodi
- uporablja pridobljena znanja iz različnih učnih predmetov v okviru izobraževanja,
- preverja sestavljena vezja še preden jih priključi na napajalno napetos za čim manjše možnosti uničenja elementov zaradi napačnih priključitev,
- se uči postopnega reševanja problemov z oblikovanjem parcialnih rešitev,
- se seznanja z načinom pretvorb fizikalnih veličin v električne in načinom prilagajanja njihovih vrednosti za nadaljno računalniško obdelavo,
- se zna povezati s strežnikom in dopolniti oz. spremeniti programe za komunikacijo med krmilniki in oddaljenimi računalniki,
- izdela celotno podatkovno verigo od senzorjev do spletnih strani in v obratni smeri do izvršilnih členov,
- upošteva pravila o varnosti in zdravju pri delu ter varovanju okolja,
- zna predstaviti rezultate svojega dela in se povezovati v timsko delo,
- sledi najnovejšim trendom na področju računalništva in informatike in jih vključuje v svoje delo.

3. Poklicne kompetence:

1. Povezovanje merilnih naprav v internetno okolje z možnostmi njihovega daljinskega upravljanja
2. Programiranje elektronskih naprav in razvoj programske opreme v različnih programskih jezikih
3. Sestavljanje enostavnih elektronskih sklopov za potrebe zajemanja podatkov in nadaljnje računalniške obdelave.
4. Nameščanje že razvite programske opreme.

4. Operativni cilji:

1. Izdelava ojačevalnika s temperaturnim senzorjem

Informativni cilji	**Formativni cilji**
Dijak: • razlikuje med električnim potencialom in napetostjo, • se spozna z negativno in pozitivno povratno vezavo, • spozna metode poenostavljanja z vpeljavo idealnih komponent, • razlikuje med invertirajočimi in neinvertirajočim ojačevalnikom, • spozna metode za odpravljanje odbojev na liniji (Schmittov trigger), • pozna način pretvorbe digitalne vrednosti v analogno, • pozna problem meritev fizikalnih veličin pasivnih elementov z operacijskim ojačevalnikom z enojnim napajanjem, • se spozna z delovanjem omejilnika napetosti in napetostnega sledilnika	Dijak: • razume osnove ohmovega zakona in zna izračunati napetosti pri različnih vezavah uporovnih elementov, • zna narisati elektronsko vezje invertiranega ali neinvertiranega ojačevalnika z uporabo operacijskega ojačevalnika, • pri načrtovanju upošteva neskončno diferencialno upornost ojačevalnika, neskončno ojačanje in diferencialno napetost nič pri negativni povratni vezavi, • uporabi primerne tehnike za pretvorbo D/A pretvorbe z operacijskim ojačevalnikom, • razume delovanje ojačevalnika z večimi operacijskimi ojačevalniki in mostično vezavo na vhodu, • zna izračunati ojačanje celotnega sistema in izbrati prave elemente za prilagoditev izhodnega signala na analogni vhod krmilnika, • uporabi omejilnik napetosti, če je potrebno in razume pomen tokovnega ojačanja pri napetostnem ojačanju 1.

2. Kodiranje algoritmov za krmiljenje elektronskih naprav za prikaz rezultatov in upravljanje izvršilnih členov

Informativni cilji	Formativni cilji
Dijak: • razlikuje med sedem segmentnim prikazovalnikom s skupno katodo in skupno anodo, • pozna način priključevanja tipk na industrijski krmilnik, • uporablja izhode za krmiljenje sedem segmentnega prikazovalnika, • vključuje tranzistorje za možnost časovnega multipleksiranja, • pozna lastnosti potenciometra in način priključitve na analogni vhod.	Dijak: • zapisuje vrednosti na sedem segmentni prikazovalnik • napiše programsko kodo dekodirnika za pretvorbo števk v ustrezno digitalno kombinacijo za krmiljenje segmentov sedem segmentnega prikazovalnika, • spozna osnove koračnega programiranja in prebira spremembo stanja tipk iz logične ena v nič in obratno, • programsko vključuje in izključuje posamezne števke na prikazovalniku, • napiše kodo generatorja pravokotnih impulzov z uporabo sistemskega časovnika, • uporablja tehniko spremembe signalov časovnika za časovno multipleksiranje vpisovanja števk na sedem segmentni prikazovalnik, ki ima vzporedno vezavo posameznih segmentov, • kombinira prebrane vrednosti iz potenciometra z ustreznim vpisom na prikazovalnik in izvršilne člene

3. Povezava krmilnika s strežnikom

Informativni cilji	**Formativni cilji**
Dijak: • pozna način komunikacije za prenos podatkov iz krmilnika na strežnik, • pozna protokol za izmenjavo podatkov, • razlikuje med asinhronim in sinhronim načinom prenosa podatkov, • primerja uporabo različnih programskih modulov pri namestitvi strežnika.	Dijak: • izbere primeren način združevanja podatkov za prenos iz krmilnika na strežnik, • prilagodi programsko kodo glede na protokol komunikacije med krmilnikom in strežnikom (USB protokol za krmilnik Arduino in TCP/IP protokol za industrijski krmilnik SIEMENS, • konfigurira podomrežja za komunikacijo z večimi krmilniki, • namešča ustrezne komponente strežnika za različen način komuniciranja s krmilniki in prilagaja programsko kodo za nemoten sprejem in oddajo, • usklajuje prenosne poti z nastavitvami ustreznih portov in s pomočjo dogodkov na sprejem sporočil komunicira s spletnimi stranmi.

4. Pisanje mini SCADE (Supervisory Control And Data Acquisition)

Informativni cilji	Formativni cilji
Dijak: • pozna in uporabi gradnike za dostop do sprejetih podatkov na spletni strani v paketni obliki, • uporablja način dogodkovnega programiranja, • arhivira sprejete podatke, • uporablja grafične in tabelarične metode za prikaz trenutnih vrednosti, • kreira vnosna polja za želene vrednosti regulacijskega sistema in piše ustrezno programsko kodo za prenos ukazov preko strežnika in krmilnika do izvršilnih členov.	Dijak: • izbira primerne načine za cepitev podatkovnih paketov, • uporablja že napisane programske bloke, pridobljene iz interneta, za grafični prikaz sprejetih vrednosti, • uporablja sistem za upravljanje različic znotraj urejevalnikov in integriranih razvojnih orodij, • testira delovanje napisanih programov in odkriva napake, • odpravlja morebitne kolizije pri pretokih podatkov v smereh sprejema iz strežnika, nazaj na strežnik in vpisovanje v bazo podatkov, • izdela enostavne in sestavljene menije.

5. Didaktična priporočila za izvedbo modula:

1. Uporaba praktičnih primerov in projektnega učenja
- Dijaki naj spoznajo osnovno zgradbo računalnika in spominske enote preko razstavljenih komponent ali virtualnih simulacij.
- Razlago vrst in lastnosti vodil lahko podkrepimo s shematskimi prikazi in praktičnimi primeri povezovanja komponent.
- Pri spoznavanju programskih jezikov naj dijaki napišejo enak program v različnih jezikih (strojni jezik, zbirnik, višji programski jezik) za boljše razumevanje razlik.

2. Interaktivni pristopi k učenju
- Dijaki naj uporabljajo razvojna okolja za mikroprocesor (mikrokrmilnik) pri konkretnih nalogah, npr. utripanje LED diod ali enostavne senzorske aplikacije.
- Uporaba eksperimentalnih plošč (npr. Arduino, Raspberry Pi) omogoča aktivno učenje in razumevanje vhodno-izhodnih enot.
- Pri inicializaciji in uporabi vhodno-izhodnih enot naj dijaki rešujejo konkretne probleme, kot so merjenje temperature ali vlažnosti.

3. Razvoj algoritmičnega mišljenja
- Dijake je smiselno voditi skozi proces razgradnje problema na manjše korake in sestavljanja algoritma pred pisanjem kode.
- Programiranje naj vključuje delo s podprogrami in prekinitvami, pri čemer dijaki preizkušajo različne scenarije uporabe.

4. Uporaba problemskega učenja

- Reševanje problemov z uporabo mikrokrmilnikov naj vključuje konkretne projekte, kot so avtomatizacija procesov ali izdelava merilnih sistemov.
- Dijaki naj testirajo in analizirajo rešitve, da razvijejo kritično razmišljanje.

5. Povezovanje teorije s prakso

- Pri uporabi tipal in zajemanju vhodnih veličin naj dijaki izvajajo eksperimente in rezultate dokumentirajo.
- Praktične vaje naj vključujejo povezovanje mikrokrmilnika z drugimi računalniškimi sistemi ter prenos podatkov v večje sisteme.

6. Individualizacija in diferenciacija pri pouku

- Dijakom, ki hitreje napredujejo, omogočiti zahtevnejše naloge, kot so optimizacija programske kode ali komunikacija med več mikrokrmilniki.
- Za dijake, ki potrebujejo več podpore, zagotoviti več ponazoritev in postopno uvajanje kompleksnih konceptov.

7. Evaluacija in spremljanje napredka

- Redna preverjanja znanja skozi praktične naloge in projekte.
- Uporaba samoocene in medsebojnega vrednotenja za krepitev refleksije o lastnem znanju.

6. Priporočila za ocenjevanje

- Ocenjevanje naj vključuje tako teoretično preverjanje kot praktične naloge.
- Pri praktičnem delu naj se ocenjuje načrtovanje, izvedba in delovanje rešitve.
- Dijaki naj prejmejo povratne informacije, ki jim omogočajo izboljšanje svojih veščin.
- Upoštevati je treba različne stile učenja in omogočiti alternativne načine prikaza znanja.
- Praktične naloge naj bodo zasnovane tako, da spodbujajo ustvarjalnost in samostojno reševanje problemov.