

Laboratorinio darbo aprašas

Dalykas: Robotika

Klasė: Ig (9 kl.)

Darbo pavadinimas:
Judėjimo pagrindai – vaikščiojimas ir balansavimas
Darbo tikslas:
Išmokti valdyti EDBOT Mini roboto kojų ir rankų judesius, suprasti balansavimo (pusiausvyros) ypatumus ir svarbą bei gebėti pritaikyti skirtingus judėjimo parametrus siekiant stabilaus roboto judėjimo.
Darbo priemonės, įranga:
• EDBOT Mini robotas • Kompiuteris su Python „Thonny“ redaktoriumi • EDBOT Python biblioteka • USB arba „Bluetooth“ ryšio modulis roboto prijungimui • Paviršiaus (pvz., juosta ant grindų) linija testavimui.
Ugdomi dalyko pasiekimai:
Informatikos dalyko pasiekimai: B2. Naudojasi algoritmų kūrimo, programavimo kalbos konstrukcijomis, programavimo aplinkomis. B3. Kuria ir vykdo algoritmus ir programas. B4. Testuoja, derina, tobulina programas.
Darbo eigos aprašymas:
1. Kojų judesių programavimas naudojant <code>ec.set_servo_position</code>: • Mokiniai nustato konkrečias kojų servo varikliukų padėtis, kad robotas atliktų vieną žingsnį į priekį. • Koduojami keli nuoseklūs žingsniai, kad robotas eitų tiesia linija.

- Stebima, kaip svarbu sinchronizuoti dešinės ir kairės kojos judesius bei išlaikyti pusiausvyrą.

2. Standartinių judesių naudojimas su `ec.set_motion`:

- Pritaikoma EDBOT bibliotekos iš anksto sukurta vaikščiojimo funkcija.
- Robotas užprogramuojamas nueiti į priekį, pasisukti 90° kampu ir vėl eiti į priekį.
- Aptariamas skirtumas tarp rankinio servo valdymo ir paruoštų judesių.

3. Ciklų naudojimas judėjimo programavimui:

- Sukuriama programa, kurioje ciklas (`for` arba `while`) kartoja roboto ėjimo veiksmus kelis kartus.
- Robotas atlieka užduotį automatiškai.
- Mokiniai keičia ciklo kartojimų skaičių ir analizuoja roboto elgseną.

Saugos reikalavimai (jei reikia):

Nedėti nieko tarp roboto judančių dalių.
Dirbti ant stabilaus paviršiaus (grindų), kad robotas nenukristų nuo stalo.
Prieš paleidžiant judesius įsitikinti, kad robotas yra pradinėje padėtyje.

Rezultatų analizė (jei reikia):

Mokiniai apgalvoja ir atsako:

- Kokias Python programos struktūras (funkcijas, ciklus) pritaikėte?
- Kaip nustatė servo varikliukų padėtis, kad robotas išlaikytų pusiausvyrą?
- Koks skirtumas tarp judėjimo programavimo servo komandomis ir `set_motion` funkcijų?
- Kaip ciklai padėjo supaprastinti kodą?

Refleksija ir vertinimas:

Su mokiniais aptarti:

- Ar pavyko robotui išlaikyti tiesią trajektoriją?
- Kuri darbo dalis buvo sudėtingiausia — servo valdymas, `set_motion` ar ciklai?
- Kaip patobulintumėte judesius, kad jie būtų sklandesni?

Vertinimo kriterijai:

Sukurtas veikiančias servo judesių kodas (2 tšk.).

Teisingai pritaikyta `set_motion` funkcija (2 tšk.).

Sėkmingai naudotas ciklas judėjimui (2 tšk.).
Kodas aiškus, su komentarais (2 tšk.).
Robotas atliko užduotį be kritimų ar klaidų (2 tšk.).

Priedai (lentelės, schemos, mokinių darbo aprašų šablonai ir t.t.):

Priedas Nr 1. Standartinių judesių žinynas

Priedas Nr 2. EdBot Mini roboto servų anatomija

Priedas Nr. 1 Standartinių judesių žinynas

Category	Motion id	Motion
Basic	3	crouch
Basic	2	get up
Basic	1	initial position
Basic	39	run forwards
Basic	18	sidestep left
Basic	17	sidestep right
Basic	4	stand
Basic	16	turn left
Basic	15	turn right
Basic	20	walk backwards
Basic	19	walk forwards
Dance	40	break dance
Dance	41	break dance flip
Dance	42	gangnam
Fight	25	karate left 1
Fight	27	karate left 2
Fight	26	karate right 1
Fight	28	karate right 2
Fight	11	left hook
Fight	10	left jab
Fight	12	left uppercut
Fight	29	push
Fight	8	right hook
Fight	7	right jab
Fight	9	right uppercut
Greet	5	bow 1
Greet	6	bow 2
Greet	14	left wave
Greet	13	right wave
Gym	24	backward roll
Gym	21	forward roll
Gym	38	head stand
Gym	23	push up
Gym	22	sit up
Sport	34	goalie block
Sport	35	goalie left
Sport	36	goalie right
Sport	37	goalie spread
Sport	30	left kick
Sport	32	left side kick
Sport	31	right kick
Sport	33	right side kick

Priedas Nr. 2 EdBot Mini roboto servų anatomija

