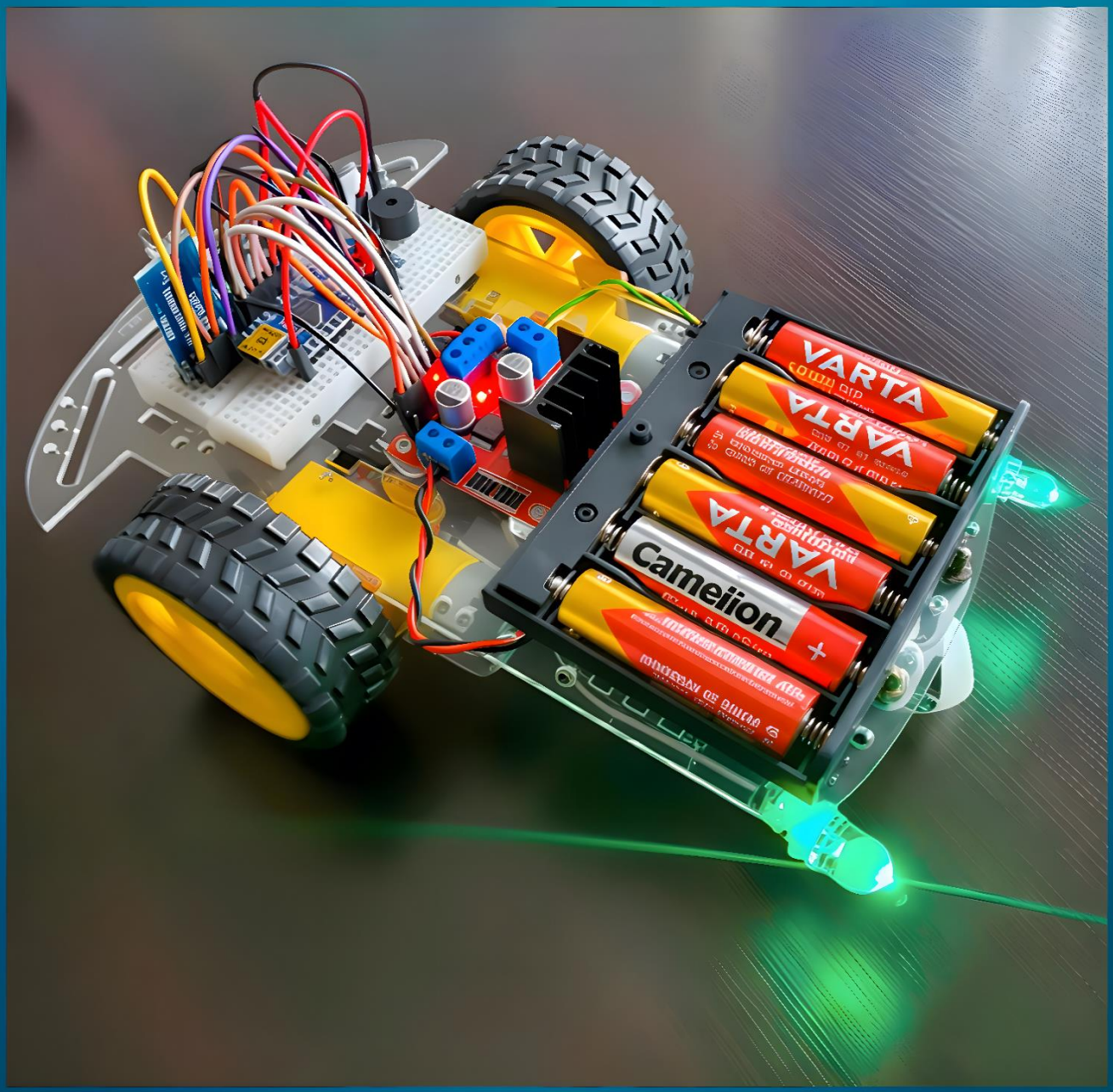
არდუინოს შემსწავლელი სახელმძღვანელო

რობოტი მანქანები



მესამე ნაწილი

არდუინოს შემსწავლელი სახელმძღვანელო

რობოტი მანქანები

III ნაწილი

სსიპ მასწავლებელთა პროფესიული განვითარების
ეროვნული ცენტრი

სახელმძღვანელო მოამზადა
მაია ნაკაშიძემ

კონსულტანტები
კახა ჟღენტი
ნუგზარ მოსულიშვილი
თეკლა პატარიძე

კორექტურა
რედაქტირება
დიზაინი
მაია ნაკაშიძე

წინასიტყვაობა

თანამედროვე მსოფლიოში რობოტიკა სულ უფრო მნიშვნელოვან როლს თამაშობს. ის გადაწყვეტილებებს გვთავაზობს უამრავ სფეროში, მედიცინიდან და მრეწველობიდან დაწყებული, განათლებით და ყოველდღიური ცხოვრების გამარტივებით დამთავრებული. რობოტიკა არა მხოლოდ ტექნოლოგიური პროგრესის სიმბოლოა, არამედ, ის ხელს უწყობს ადამიანის შესაძლებლობების გაფართოებას, ავტომატიზაციასა და ინოვაციებს, რაც ცვლის ჩვენს გარემოს.

ამ წიგნში გაგაცნობთ ჭკვიან მანქანებს/რობოტებს, რომლებიც დაფუძნებულია 2WD (ორბორბლიანი წამყვანი) და 4WD (ოთხბორბლიანი წამყვანი) მოდელებზე. ეს პროექტები არა მხოლოდ პრაქტიკული გამოცდილების მიღების საშუალებაა, არამედ, ისინი გვიჩვენებენ, თუ როგორ შეიძლება რობოტიკის პრინციპების გამოყენება რეალური პრობლემების გადასაჭრელად. თითოეული პროექტი დეტალურად არის აღწერილი, დაწყებული კონცეფციიდან და დამთავრებული მისი რეალიზაციით, რათა მკითხველმა შეძლოს ყველა ნაბიჯის გააზრება და გამეორება.

რობოტიკის პროექტები, როგორიცაა 2WD და 4WD მანქანები, განსაკუთრებით აქტუალურია დღეს. ისინი გვეხმარებიან გავიგოთ, თუ როგორ მუშაობს ავტონომიური სისტემები, როგორ ხდება მონაცემთა დამუშავება და როგორ შეიძლება მექანიკური კომპონენტების ინტეგრირება ელექტრონულ და პროგრამულ ნაწილებთან. ეს ცოდნა აუცილებელია არა მხოლოდ ინჟინრებისთვის, არამედ ყველასთვის, ვინც ინტერესდება მომავლის ტექნოლოგიებით.

თანამედროვე ცხოვრებაში რობოტიკის გამოწვევები მრავალფეროვანია. ის მოიცავს ავტონომიური მანქანების განვითარებას, ჭკვიანი სახლების სისტემების დანერგვას, მედიცინაში რობოტების გამოყენებას და ინდუსტრიაში ავტომატიზაციის პროცესების გაუმჯობესებას. ამ გამოწვევებს თან ახლავს ტექნიკური, ეთიკური და სოციალური საკითხები, რომლებიც საჭიროებენ ღრმა გაგებასა და შემოქმედებით მიდგომას.

ამ წიგნის მიზანია არა მხოლოდ პრაქტიკული ცოდნის მიწოდება რობოტიკით დაინტერესებული პირებისთვის, არამედ მკითხველის შთაგონება, რათა მან თავად შეძლოს ინოვაციური პროექტების შექმნა. ვიმედოვნებთ, რომ ეს მასალა გახდება თქვენი შემოქმედებითი მოგზაურობის საწყისი წერტილი რობოტიკის სამყაროში.

არდუინოს შემსწავლელი სახელმძღვანელო (I და II ნაწილი)

არდუინოს შემსწავლელი სახელმძღვანელო - რობოტი მანქანები (III ნაწილი)

ბიბლიოთეკები და კოდები

ელექტრონული ვერსია, საჭირო ბიბლიოთეკები და პროექტების კოდები შეგიძლიათ იხილოთ:

ვებგვერდი: https://chkhirkedela2019.wixsite.com/mysite

გვერდი ARDUINO UNO

ასევე, შეგიძლიათ იხილოთ ვებმისამართებზე:

- არდუინოს შემსწავლელი სახელმძღვანელო - რობოტი მანქანები, პროექტების კოდები: https://bit.ly/3Y7044K
- არდუინოს შემსწავლელი სახელმძღვანელო (I ნაწილი) - პროექტების კოდები: https://bit.ly/3NoeyYv
- არდუინოს შემსწავლელი სახელმძღვანელო (II ნაწილი) - პროექტების კოდები: https://bit.ly/4i5gGIY
- ბიბლიოთეკები: https://bit.ly/413fjht

ასევე, შეცდომების გასწორება იხილეთ:

ვებგვერდი: https://chkhirkedela2019.wixsite.com/mysite

გვერდი ARDUINO UNO

რაჭომ რობოჭი მანქანები?!

თანამედროვე ტექნოლოგიების ეპოქაში, სადაც ხელოვნური ინტელექტი და ავტომატიზაცია ჩვენი ყოველდღიურობის განუყოფელი ნაწილი გახდა, რობოტ მანქანებს განსაკუთრებული ადგილი უჭირავს. ისინი არა მხოლოდ სამეცნიერო ფანტასტიკის ჟანრის გმირები არიან, არამედ იდეალური საგანმანათლებლო ინსტრუმენტებია, რომლებიც აერთიანებენ თეორიას პრაქტიკასთან, აბსტრაქტულ კოდს – ფიზიკურ მოქმედებასთან.

ამ წიგნის შექმნის იდეა სწორედ ამ სინთეზიდან წარმოიშვა. „რობოტი მანქანების“ სამყარო გაცილებით ღრმა და მრავალფეროვანია, ვიდრე ერთი კონკრეტული ფუნქციის შესრულება. ეს არის მიკროკოსმოსი, სადაც მექანიკა ხვდება ელექტრონიკას, პროგრამირება – ლოგიკას და კრეატიულობა – ინჟინერიას.

რას გიზიარებთ ეს წიგნი?

ამ გზამკვლევში ჩვენ გადავხედავთ რობოტი მანქანების შექმნის სრულ ციკლს – მარტივი 2WD (ორბორბლიანი წამყვანი) და 4WD (ოთხბორბლიანი წამყვანი) მანქანებიდან დაწყებული, სრულფასოვან, ავტონომიურ და დისტანციურად მართვად მანქანებამდე. ჩვენი მოგზაურობა მოიცავს:

ძირითადი აღქმა: ინფრაწითელი (IR) სენსორების გამოყენებით ჩვენ ვასწავლით რობოტს „დაინახოს“ შავი ხაზი და მიჰყვეს მას უწყვეტად, რაც კლასიკური, მაგრამ უაღრესად ეფექტური ალგორითმია ნავიგაციისთვის.

გარემოსთან ურთიერთქმედება: ულტრაბგერითი სენსორის დახმარებით, რობოტი შეძლებს „გაეცნოს“ გარშემო სივრცეს, აღმოაჩინოს დაბრკოლებები და აუაროს მათ გვერდი, რაც ავტონომიური მანქანების მუშაობის ბაზისს წარმოადგენს.

დისტანციური კონტროლი: ჩვენ გამოვიყენებთ **Wi-Fi**, **Bluetooth** და **NRF24L01** მოდულების სიმძლავრეს, რათა რობოტი ჩვენი სმარტფონის ან სპეციალური პულტის გამოყენებით ვმართოთ და თავი ავარიდოთ მართვის კაბელებს.

ძრავის მართვა: L298N Driver-ის მეშვეობით ჩვენ ვისწავლით, თუ როგორ უნდა ვაკონტროლოთ ძრავების სიჩქარე და მიმართულება, რაც არის ნებისმიერი მოძრაობის საფუძველი.

ბალანსის ხელოვნება: ჩვენ განვიხილავთ, თუ როგორ გადააქცევს **GY-521 MPU6050** აჩქარების/გიროსკოპის სენსორი ჩვეულებრივ 2WD მანქანას თვითბალანსირებად

სისტემად, აღწევს სეგვეის მუშაობის პრინციპს – ეს არის ერთ-ერთი ყველაზე დამაჯერებელი დემონსტრაცია სენსორებისა და ალგორითმების ძალისა.

მანიპულირება: წიგნის ერთ-ერთი ყველაზე მოხდენილი ნაწილია **მექანიკური მკლავის** ინტეგრირება. ჩვენ ვისწავლით, თუ როგორ შეუძლია რობოტს, შავ ზოლზე მოძრაობის პროცესში ასწიოს საგანი, გადადოს გვერდზე და განაგრძოს ფუნქციონირება – რაც ახლოსაა ინდუსტრიულ რობოტებთან და ლოჯისტიკურ სისტემებთან.

რაზომ არის ეს ყველაფერი მნიშვნელოვანი?

რობოტი მანქანების აწყობა და პროგრამირება – ეს არ არის მხოლოდ საჭირო სენსორების შეგროვება. ეს არის პრობლემების გადაჭრის, კრიტიკული აზროვნების და მოთმინების სკოლა. თითოეული დაკავშირებული გამტარი, თითოეული დაწერილი კოდის სტრიქონი და თითოეული წარმატებით შესრულებული დავალება გვასწავლის ფუნდამენტურ პრინციპებს:

სენსორების ლოგიკა: როგორ იქცევა ფიზიკური სამყაროს სიგნალი ციფრულ ინფორმაციად.

სისტემების ინტეგრაცია: როგორ მუშაობს ერთად სხვადასხვა ტექნოლოგია (მექანიკა, ელექტრონიკა, პროგრამული უზრუნველყოფა).

ალგორითმული აზროვნება: როგორ დავყოთ კომპლექსური ამოცანა (მაგ., „აიღე ობიექტი“) მარტივ, თანმიმდევრულ ნაბიჯებად.

ეს წიგნი გვაძლევს ცოდნას და გვივითარებს საჭირო უნარებს, რათა შევძლოთ არა მხოლოდ ავაწყოთ რობოტი, არამედ გავიგოთ, რატომ მუშაობს ის ისე, როგორც მუშაობს. ეს არის გზის დასაწყისი ავტომატიზაციის, რობოტოტექნიკისა და ხელოვნური ინტელექტის საოცარ სამყაროში.

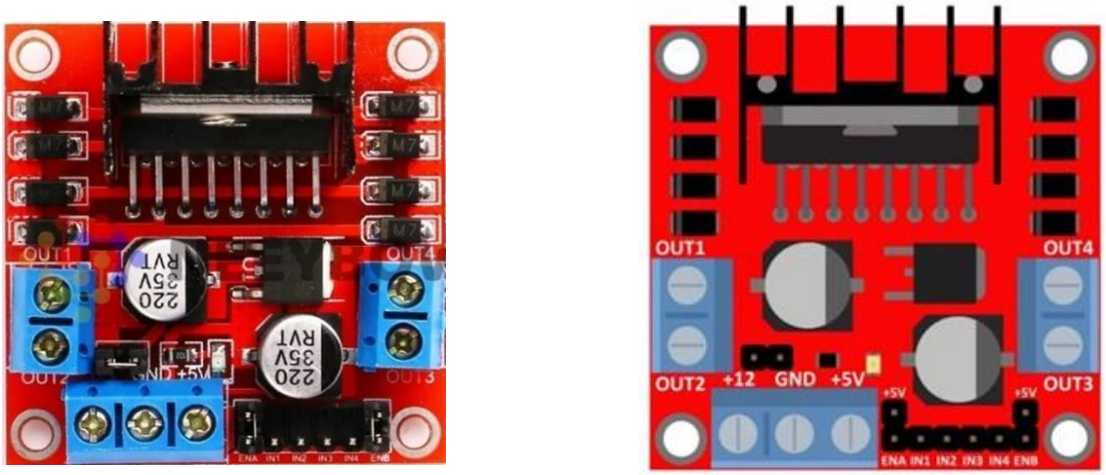
მოდით, ერთად ჩავუღრმავდეთ ამ საოცარ სამყაროს და გავაკეთოთ მანქანები, რომლებიც არა მხოლოდ მოძრაობენ, არამედ „აზროვნებენ“ და ურთიერთქმედებენ გარემოსთან.

სარჩევი

როგორ ავაწყოთ რობოტი მანქანის ძარა

ამ თავში ჩვენ შემოგთავაზებთ რობოტი მანქანის ძარას აწყობის ნაბიჯ-ნაბიჯ ინსტრუქციას და იმ კომპონენტებს, რომლებიც წიგნში წარმოდგენილ ყველა პროექტში დაგჭირდებათ.

ძრავას დრაივერი L298N, რომელიც მართავს რედუქტორულ ძრავებს. მათი საშუალებით რობოტი მანქანა გადაადგილდება წინ და უკან, მოტრიალდება მარჯვნივ და მარცხნივ (იხ. სურათი 1). ამ კომპონენტის შესახებ სრულ ინფორმაციას შემოგთავაზებთ ქვემოთ.



სურათი 1. ძრავას დრაივერი L298N

ასევე, გიჩვენებთ, როგორ უნდა დააკავშიროთ კვების წყაროსთან ჩამრთველ-გამომრთველი (იხ. სურათი 2), რაც უფრო მოსახერხებელს გახდის თქვენთვის მანქანის „დაქოქვა“-„გაჩერებას“.

სურათი 2. ჩამრთველ-გამომრთველი

ამას გარდა, თქვენ დაგჭირდებათ სარჩილავი და მისი კომპონენტები: კალა, ფლუსი (კანიფოლი) და სარჩილავის საწმენდი (იხ. სურათი 3).

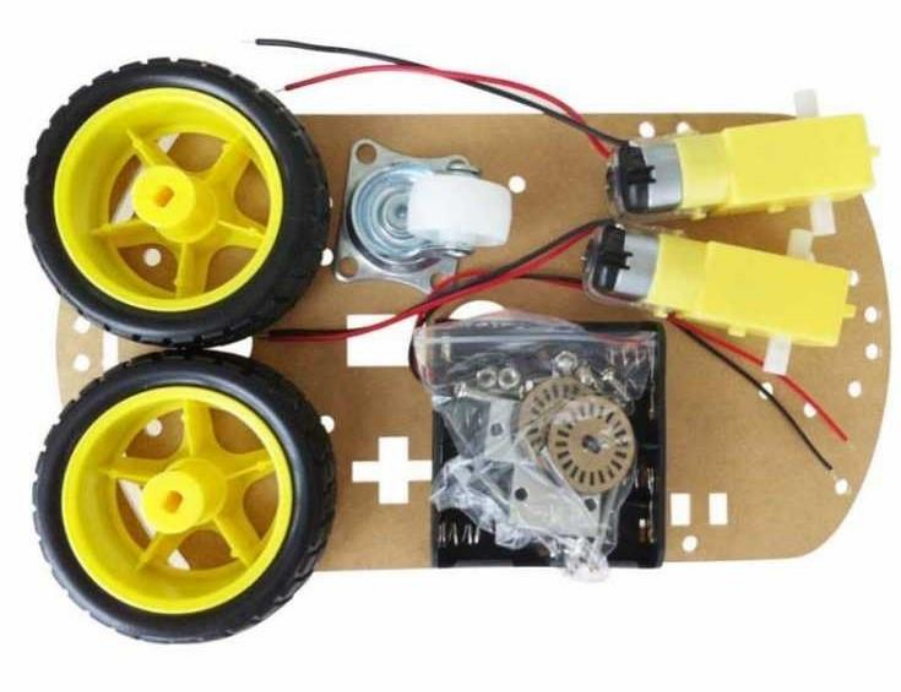
სურათი 3. სარჩილავი, კალა, ფლუსი და საწმენდი

ასევე, ელემენტების ბუდის (და ნებისმიერი სხვა საჭირო კომპონენტების) დასამაგრებლად შეგიძლიათ გამოიყენოთ ორმაგი წებოვანი ქაღალდი (სკოჩი) ან ცხელი წებო (იხ. სურათი 4).

სურათი 4. ცხელი წებო

ახლა კი შეგვიძლია დავიწყოთ 2WD (ორბორბლიანი წამყვანი) რობოტი მანქანის ძარის აწყობა (ანალოგიურად ეწყობა 4WD მანქანაც):

თავდაპირველად, გახსენით მანქანის კომპლექტი და ყველა დეტალს გაეცანით ყურადღებით. აკრილის დეტალებიდან, სურვილის შემთხვევაში, შეგიძლიათ მოხსნათ დამცავი ფირები.

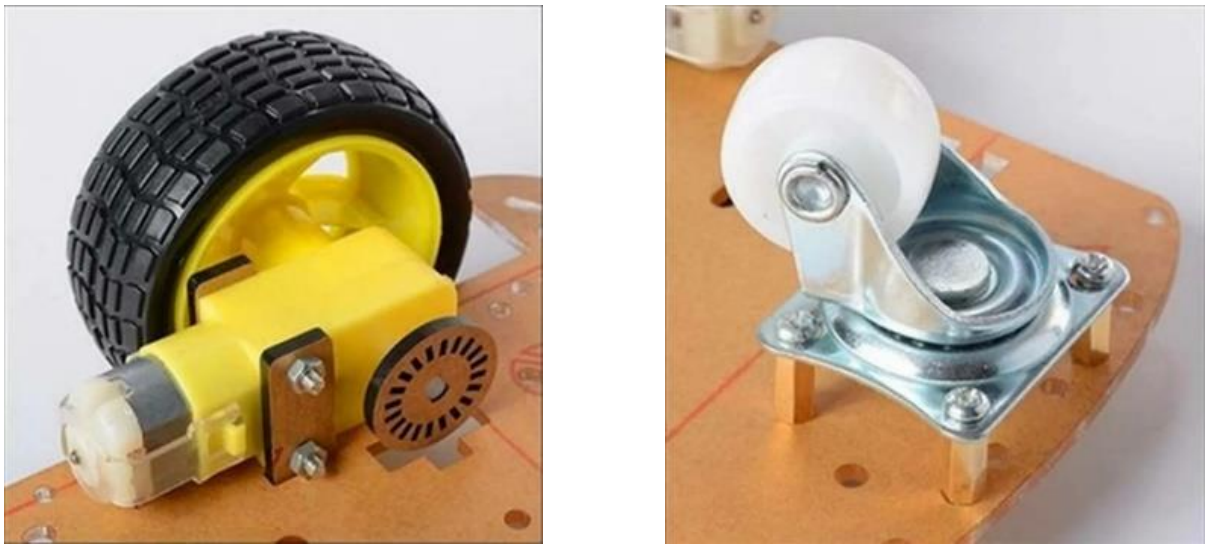


სურათი 5. 2WD მანქანის კომპლექტი

2. ძრავებს მირჩილვის წესით დაამაგრეთ გამტარები (იხ. სურათი 6).

სურათი 6.

3. დაამაგრეთ ძრავები და წინა ბორბალი ძირითად დაფაზე (იხ. სურათი 7)

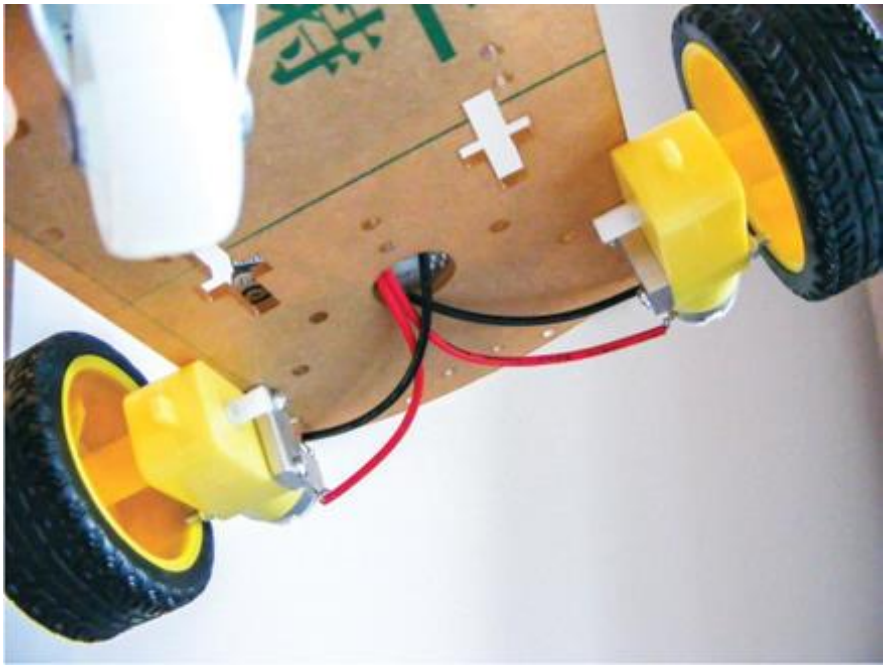


სურათი 7.

ასე უნდა გამოიყურებოდეს თქვენ მიერ აწყობილი 2WD მანქანა (იხ. სურათი 8).

სურათი 8.

ასე გამოიყურება ქვემოდან ძრავების გამტარების პოზიცია დრაივერთან დასაკავშირებლად (იხ. სურათი 9).

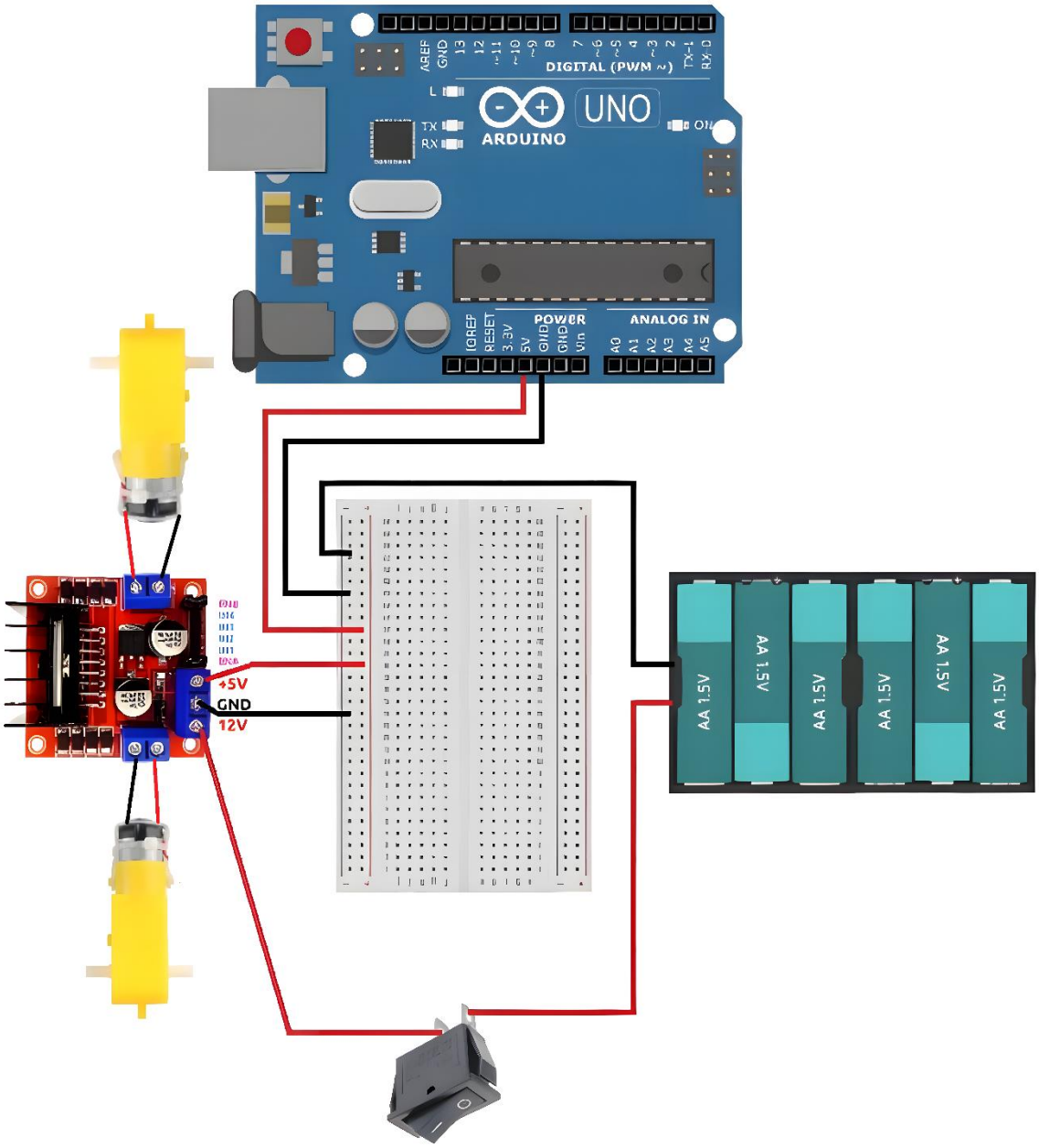


სურათი 9.

4. კვების წყაროდ შეგიძლიათ გამოიყენოთ 1.5 v-იანი ან 3.7 v-იანი ლითიუმის ელემენტი. ეს დამოკიდებულია კონკრეტული პროექტისთვის საჭირო კომპონენტის სიმძლავრეზე (იხ. სურათი 10).

სურათი 10. ელემენტების ბუდეები

ახლა ვნახოთ, როგორ უნდა დავაკავშიროთ სამაკეტო დაფის საშუალებით ერთმანეთთან არდუინო, ძრავას დრაივერი L298N, DC ძრავები, კვების წყარო და ჩამრთველი (იხ. სურათი 11).



სურათი 11

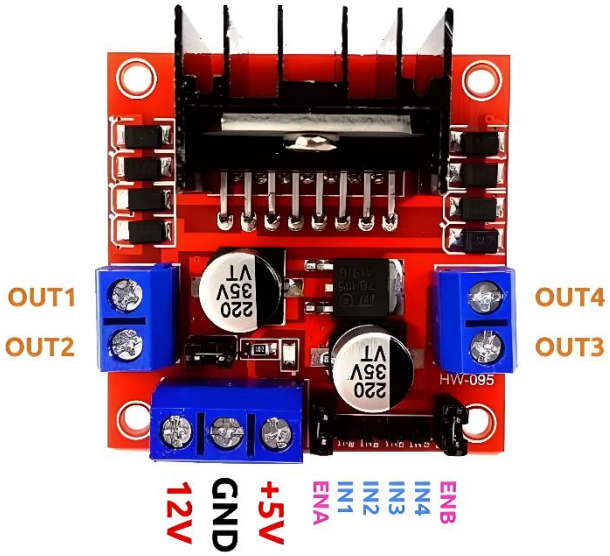
კვების წყაროდ, როგორც ზემოთ უკვე გითხარით, შეგიძლიათ გამოიყენოთ 1.5v-იანი ან 3.7 v-იანი ლითიუმის ელემენტი. აკუმულატორებისთვის შეგიძლიათ შეიძინოთ ბუდე ჩამრთველი-გამომრთველით და პირდაპირ დააკავშიროთ ძრავას დრაივერთან.

ძრავას დრაივერი L298N

L298N ძრავას დრაივერის მუშაობის პრინციპი და საკონტაქტო პინების განმარტება

L298N არის ორმაგი H-ხიდის ინტეგრირებული სქემა, რომელიც შესაძლებელს ხდის DC ძრავებისა და ბიჯური ძრავების მართვას. ის ფართოდ გამოიყენება რობოტი მანქანების პროექტებში. ის მართავს რედუქტორულ ძრავებს. მათი საშუალებით რობოტი მანქანა გადაადგილდება წინ და უკან, მოტრიალდება მარჯვნივ და მარცხნივ (იხ. სურათი 12).

- **საკონტაქტო პინების განმარტება:**



სურათი 12. ძრავას დრაივერი L298N

კვების პინები:

1. **VCC (12V Input)**: ძრავას ელექტრომომარაგება
2. **GND**: მიწა (დაკავშირებული უნდა იყოს Arduino-ს GND-თან)
3. **+5V Output**: 5V გამომავალი (შეიძლება გამოყენებულ იქნას Arduino-სთვის)

ძრავის კონტროლის პინები:

1. **ENA (Enable A)**: PWM სიგნალი ძრავას A სიჩქარის კონტროლისთვის
2. **IN1, IN2**: ლოგიკური სიგნალები ძრავას A მიმართულების კონტროლისთვის
3. **ENB (Enable B)**: PWM სიგნალი ძრავას B სიჩქარის კონტროლისთვის
4. **IN3, IN4**: ლოგიკური სიგნალები ძრავას B მიმართულების კონტროლისთვის

ძრავის გამოსასვლელები:

1. **OUT1, OUT2**: A ძრავას კონექტორები
2. **OUT3, OUT4**: B ძრავას კონექტორები

რობოტი მანქანის პროექტებში გამოყენების მაგალითი

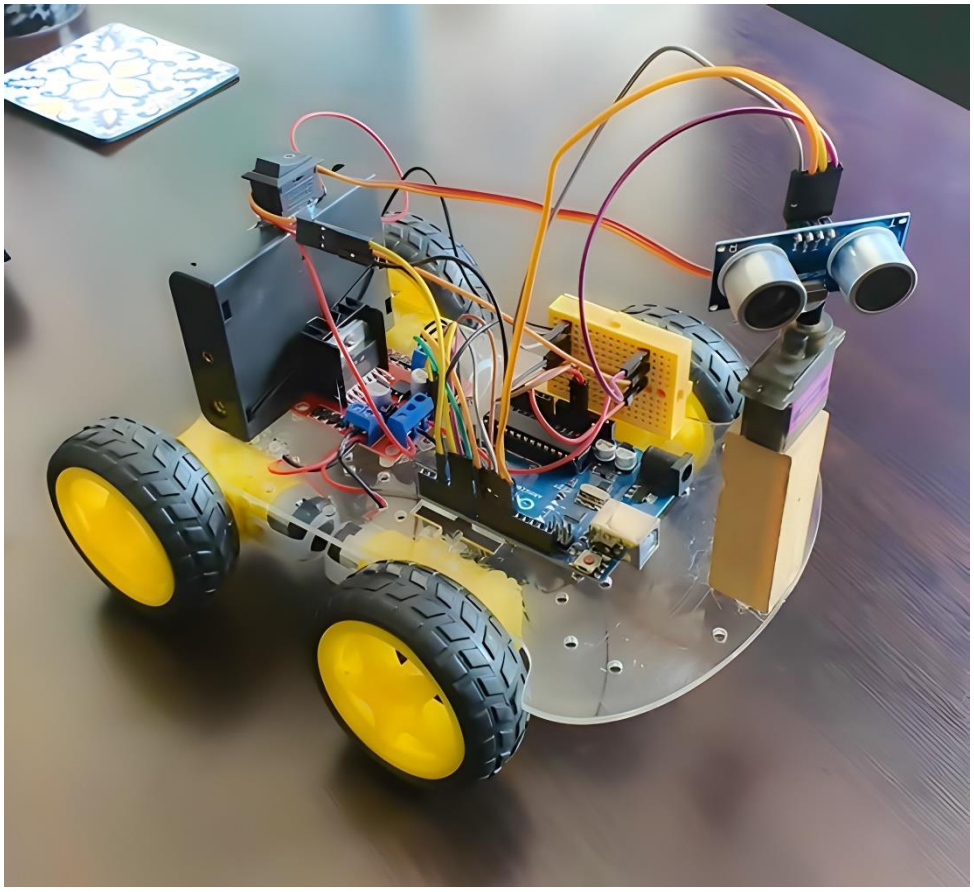
1. **მიმართულების კონტროლი**:
    - IN1=HIGH, IN2=LOW → ძრავა ბრუნავს წინ
    - IN1=LOW, IN2=HIGH → ძრავა ბრუნავს უკან
    - IN1=IN2 → ძრავა გაჩერებულია
2. **სიჩქარის კონტროლი**:
    - PWM სიგნალის გამოყენება ENA/ENB პინებზე (0-255 მნიშვნელობები Arduino-ში)

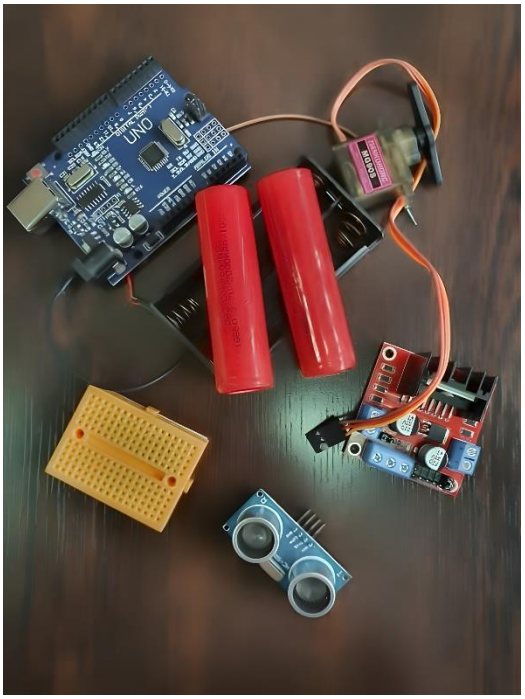
L298N არის საიმედო და მარტივად გამოსაყენებელი კომპონენტი რობოტი მანქანების პროექტებში DC ძრავების მართვისთვის.

გაითვალისწინეთ!

იმ შემთხვევაში, თუ DC ძრავები სხვადასხვა მიმართულებით ბრუნავენ, შეგიძლიათ გადაადგილოთ გამტარები ძრავას დრაივერზე (იხ. სქემა, სურათი 12) ან კოდში შეიტანოთ ცვლილება.

რობოტი მანქანა, რომელიც გვეხმარება უვლის დაბრკოლებებს





საჭირო რესურსი

- 4WD მანქანის კომპლექტი
- Arduino UNO
- ძრავას დრაივერი L298N
- ულტრაბგერის სენსორი
- Servo ძრავა MG90S
- პატარა სამაკეტო დაფა
- 2 ცალი ლითიუმის აკუმულატორი 3.7 V და ბუდე
- ჩამრთველ-გამომრთველი
- გამტარები: მამალ-მამალი და დედალ-მამალი

საჭირო ბიბლიოთეკები

- Servo
- NewPing

ულტრაბგერითი სენსორის მუშაობის პრინციპი

ამ პროექტში ჩვენ გამოვიყენებთ ულტრაბგერით სენსორს და შევქმნით ჭკვიან მანქანას, რომელიც გვერდს უვლის დაბრკოლებებს.

ულტრაბგერითი სენსორი გარდაქმნის ულტრაბგერითი ტალღის სიგნალებს ელექტრულ სიგნალებად. ულტრაბგერით ტალღებს აქვთ სითხეებსა და მყარ სხეულებში არეკვლის უნარი, განსაკუთრებით, გაუმჭვირვალე მყარ ნაწილებში. როდესაც ულტრაბგერითი ტალღები ხვდება მინარევებს ან ინტერფეისებს, ისინი წარმოქმნიან მნიშვნელოვან ანარეკლს ექო-სიგნალების სახით. სენსორი მუშაობს რადარის ანალოგით: ის აგზავნის ულტრაბგერით სიგნალს. როცა ეს სიგნალი ხვდება ობიექტს, ის ექოსავით აირეკლება და დრო სიგნალსა და ექოს შორის გამოიყენება მანძილის გამოსათვლელად (იხ. სურათი 1).

სურათი 1. ულტრაბგერითი სენსორი

სენსორი შედგება ორი ძირითადი კომპონენტისგან: ერთი გამოსცემს იმპულსს (კონტაქტი Trig), ხოლო მეორე (კონტაქტი Echo) - აღმოაჩენს იმპულსს. კონკრეტულად ეს სენსორი არ გამოდგება დიდ მანძილზე სამუშაოდ. მისი მაქსიმალური მნიშვნელობა შემოიფარგლება 20-30 სანტიმეტრით. თუმცა, რა თქმა უნდა, არის გაცილებით ძლიერი ულტრაბგერითი სენსორებიც, მათ შორის, წყალგაუმტარი.

როგორც უკვე აღვნიშნეთ, ულტრაბგერითი სენსორი მუშაობს ბგერითი ტალღების გამოყენებით, რათა განსაზღვროს მანძილი ან აღმოაჩინოს ობიექტები. ის ფუნქციონირებს შემდეგნაირად:

1. **ხმის ტალღების გამოცემა:** სენსორი გამოსცემს ულტრაბგერით ტალღებს (ხმის ტალღები, რომლებიც ადამიანის სმენისთვის შეუმჩნეველია, ჩვეულებრივ 20 kHz-ზე მაღალი სიხშირით).

2. **ტალღების არეკვლა:** ეს ტალღები ვრცელდება ჰაერში და როდესაც ისინი ეჯახებიან ობიექტს, ირეკლება უკან, სენსორისკენ.

3. **არეკლილი ტალღების დაჭერა:** სენსორი იჭერს არეკლილ ტალღებს და ზომავს დროს, რომელიც სჭირდება ტალღებს სენსორიდან ობიექტამდე და უკან დასაბრუნებლად.

4. **მანძილის გამოთვლა:** სენსორი იყენებს ხმის სიჩქარეს ჰაერში (დაახლოებით 343 მეტრი წამში ოთახის ტემპერატურაზე) და დროს, რათა გამოთვალოს მანძილი ობიექტამდე. ფორმულა არის:

   მანძილი = ხმის სიჩქარე × დროზე და გაყოფილი 2-ზე, რადგან ტალღა გადის მანძილს ორჯერ - ობიექტამდე და უკან).

ულტრაბგერითი სენსორები გამოიყენება მრავალ სფეროში: რობოტიკაში, მედიცინაში, სამრეწველო ავტომატიზაციაში, მათ შორის, მანქანებში - პარკირების დასახმარებლად.

ქვემოთ მოცემულია სქემა (იხ. სურათი 2), რომელიც საჭიროა ასეთი რობოტის შესაქმნელად.

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino UNO
ENA	Pin 5
IN1	Pin 7
IN2	Pin 8
IN3	Pin 9
IN4	Pin 10
ENB	Pin 6

2. დააკავშირეთ ერთმანეთთან არდუინო და Servo ძრავა:

Servo ძრავა	Arduino UNO
შავი გამტარი (GND)	GND
წითელი გამტარი (5V)	5V
ყვითელი გამტარი	Pin 3

3. დააკავშირეთ ერთმანეთთან ულტრაბგერითი სენსორი და არდუინო:

ულტრაბგერითი სენსორი	Arduino UNO
VCC (5V)	5V
Trig	Pin 11
Echo	Pin 12
GND	GND

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

4. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი, Servo ძრავა და ულტრაბგერითი სენსორი (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

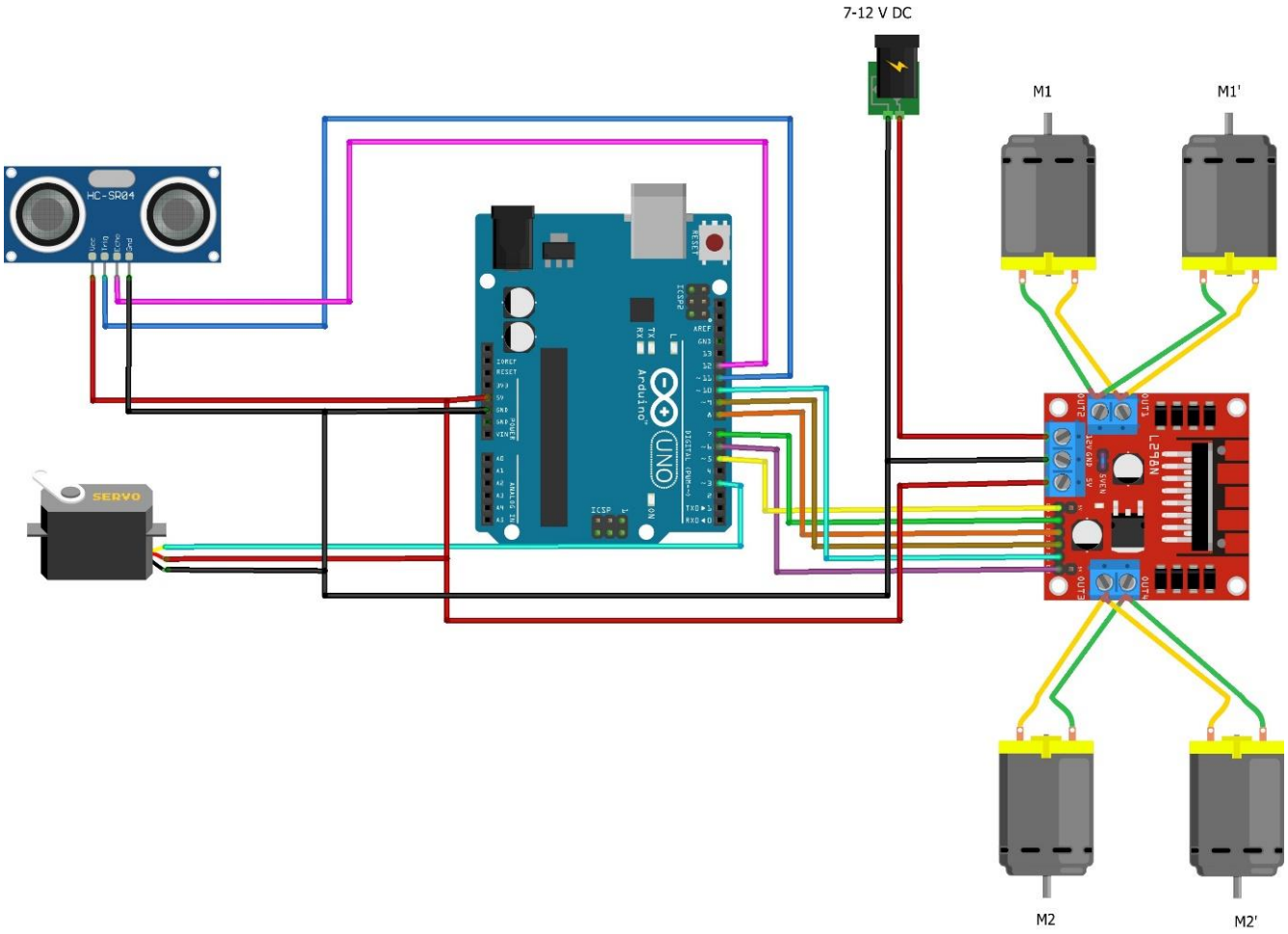
Arduino UNO	სამაკეტო დაფა
5V	+
GND	GND

ძრავას დრაივერი L298N	სამაკეტო დაფა
+5V	+
GND	GND

Servo ძრავა	სამაკეტო დაფა
წით. გამტარი	+
GND	GND

ულტრა ბგერითი სენსორი	სამაკეტო დაფა
VCC	+
GND	GND

5. დააკავშირეთ არდუინო კომპიუტერთან.



სურათი 2. სქემა

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ პორტი.

6. ჩატვირთეთ კოდი.

7. გამორთეთ არდუინო კომპიუტერიდან. ჩაალაგეთ ელემენტები ბუდეში და „დაქოქეთ“ მანქანა.

კოდის მუშაობის პრინციპი

კოდი იყენებს ულტრაბგერით სენსორს და Servo ძრავას დაბრკოლებების აღმოსაჩენად და მათთვის გვერდის ასავლელად. ქვემოთ მოცემულია კოდის მოკლე ახსნა:

1. კომპონენტები:

- ულტრაბგერითი სენსორი: გამოიყენება NewPing ბიბლიოთეკა, რომელიც ზომავს მანძილს დაბრკოლებამდე (ULTRASONIC_SENSOR_TRIG და ULTRASONIC_SENSOR_ECHO პინებზე).
- Servo ძრავა: გამოიყენება ულტრაბგერითი სენსორის მიმართულების შესაცვლელად (SERVO_PIN).
- ძრავები: რობოტს აქვს ორი ძრავა (მარცხენა და მარჯვენა), რომელიც კონტროლდება Arduino-ს პინების მეშვეობით.

2. კოდის ძირითადი ლოგიკა:

❖ **setup():**

- ინიციალიზებულია ძრავების და Servo ძრავას პინები.
- Servo ძრავა დაყენებულია ცენტრალურ პოზიციაზე (90 გრადუსი).
- რობოტის ძრავები გაჩერებულია (rotateMotor(0, 0)).

❖ **loop():**

- ყოველ ციკლში იზომება მანძილი დაბრკოლებამდე (mySensor.ping_cm()).
- თუ დაბრკოლება აღმოჩენილია 30 სმ-ის ფარგლებში (DISTANCE_TO_CHECK), რობოტი ასრულებს შემდეგ ნაბიჯებს:

1. **გაჩერება:** რობოტი ჩერდება (rotateMotor(0, 0)).
2. **უკან დახევა:** რობოტი უკან იხევს (rotateMotor(-MAX_MOTOR_ADJUST_SPEED, -MAX_MOTOR_ADJUST_SPEED)).
3. **Servo ძრავას მობრუნება მარცხნივ:** Servo ძრავა მობრუნებულია მარცხნივ (myServo.write(180)) და იზომება მანძილი მარცხნივ (distanceLeft).
4. **Servo ძრავას მობრუნება მარჯვნივ:** Servo ძრავა მობრუნებულია მარჯვნივ (myServo.write(0)) და იზომება მანძილი მარჯვნივ (distanceRight).
5. **Servo ძრავას ცენტრში დაბრუნება:** Servo ძრავა ბრუნდება ცენტრში (myServo.write(90)).
6. **მიმართულების არჩევა:** რობოტი ირჩევს მიმართულებას დაბრკოლებების გათვალისწინებით:

- თუ მარცხნივ არ არის დაბრკოლება (distanceLeft == 0), რობოტი მარცხნივ მობრუნდება.
- თუ მარჯვნივ არ არის დაბრკოლება (distanceRight == 0), რობოტი მარჯვნივ მობრუნდება.
- თუ მარცხნივ მეტი ადგილია, ვიდრე მარჯვნივ (distanceLeft >= distanceRight), რობოტი მარცხნივ მობრუნდება.
- წინააღმდეგ შემთხვევაში, რობოტი მარჯვნივ მობრუნდება.

7. **გაჩერება:** რობოტი ჩერდება (rotateMotor(0, 0)).

- თუ დაბრკოლება არ არის აღმოჩენილი, რობოტი წინ მიიწევს (rotateMotor(MAX_REGULAR_MOTOR_SPEED, MAX_REGULAR_MOTOR_SPEED)).

3. ძრავების კონტროლი:

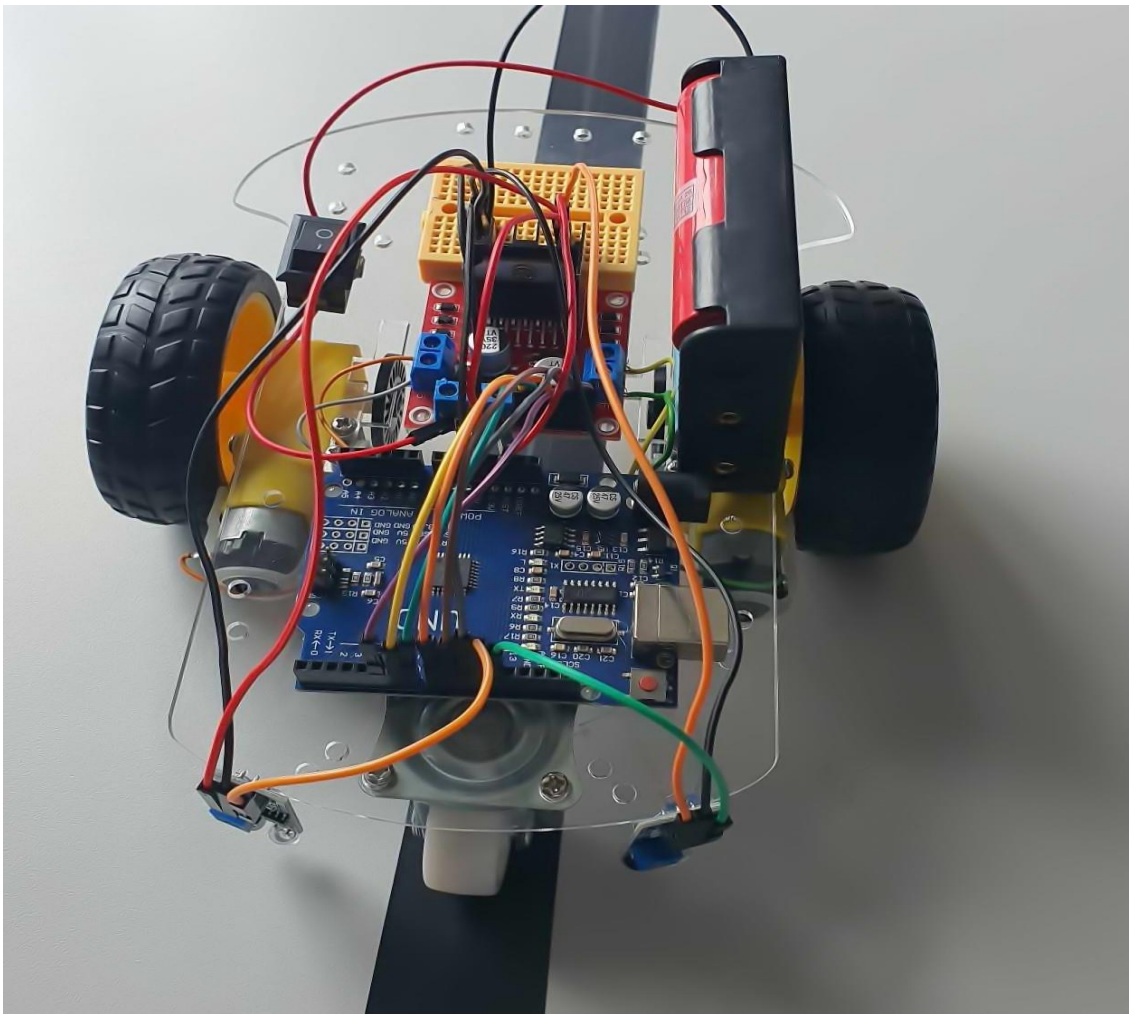
- rotateMotor(int rightMotorSpeed, int leftMotorSpeed): ეს ფუნქცია აკონტროლებს ძრავების მიმართულებას და სიჩქარეს.
  - თუ სიჩქარე დადებითია (> 0), ძრავა მოძრაობს წინ.
  - თუ სიჩქარე უარყოფითია (< 0), ძრავა მოძრაობს უკან.
  - თუ სიჩქარე ნულის ტოლია (0), ძრავა გაჩერებულია.
  - analogWrite() გამოიყენება ძრავების სიჩქარის კონტროლისთვის.

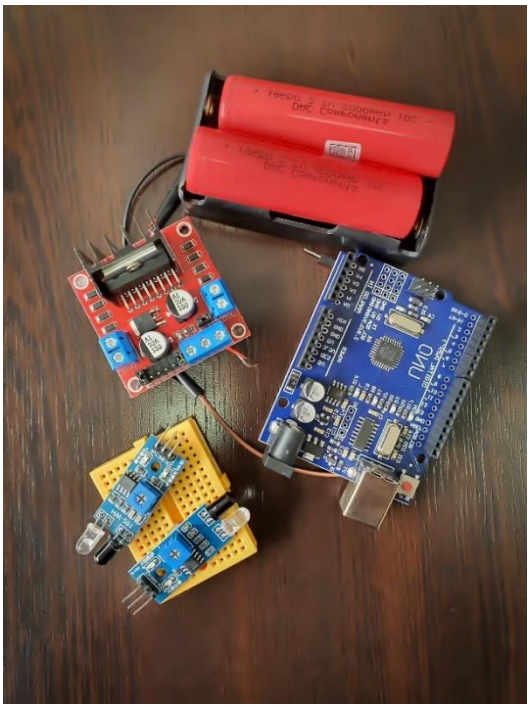
4. პარამეტრები:

- MAX_REGULAR_MOTOR_SPEED: რობოტის ნორმალური სიჩქარე (75).
- MAX_MOTOR_ADJUST_SPEED: რობოტის სიჩქარე დაბრკოლების თავიდან ასაცილებლად (150).
- DISTANCE_TO_CHECK: მანძილი, რომლის დროსაც რობოტი რეაგირებს დაბრკოლებაზე (30 სმ).

კოდი არ იყენებს მიმდევრობით კომუნიკაციას, მაგრამ შეგიძლიათ კოდში Serial.print()-ის დამატება და მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor) მანძილებსა და რობოტის მოქმედებებზე დაკვირვება.

შავ ზოლზე მოძრავი რობოტი მანქანა



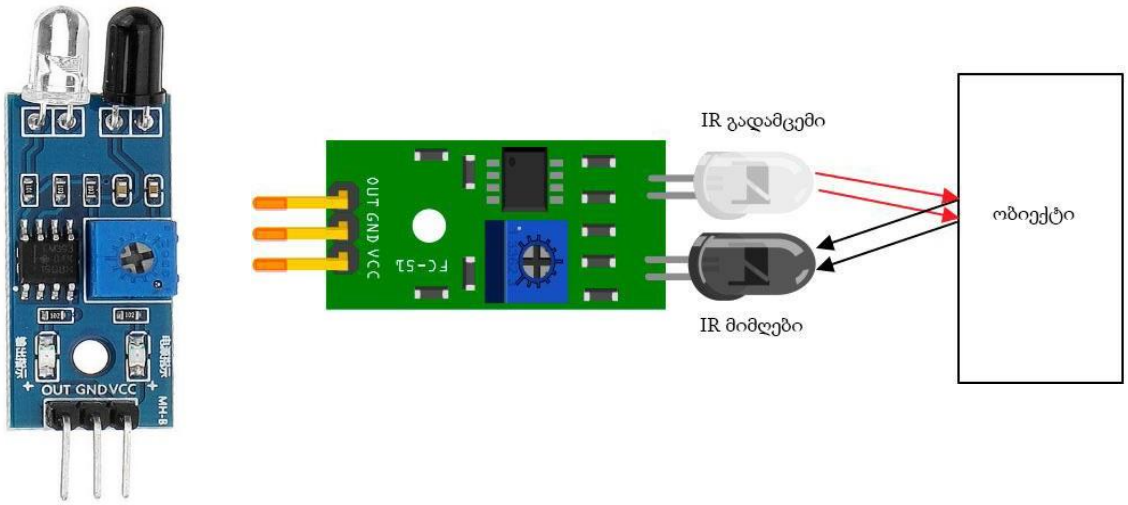


საჭირო რესურსი

- 2WD მანქანის კომპლექტი
- Arduino UNO
- ძრავას დრაივერი L298N
- 2 ცალი ინფრაწითელი (IR) სენსორის მოდული
- პატარა სამაკეტო დაფა
- 2 ცალი ლითიუმის ელემენტი 3.7 V და ბუდე
- ჩამრთველ-გამომრთველი
- გამტარები: მამალ-მამალი და დედალ-მამალი

როგორ მუშაობს ინფრაწითელი (IR) სენსორი

ეს სენსორები შეიძლება გამოყენებულ იქნას ორი ფუნქციის შესასრულებლად: შავი და თეთრი ფერების იდენტიფიკაციისთვის და ასევე, დაბრკოლებების დასაფიქსირებლად. ის განსაკუთრებით პოპულარულია სხვადასხვა სახის რობოტებთან დაკავშირებული პროექტების განსახორციელებლად, როგორიცაა, მაგალითად: რობოტი, რომელიც გვერდს უვლის დაბრკოლებას ან, ჩვენ შემთხვევაში, რობოტი, რომელიც რაიმე წინასწარ განსაზღვრულ ხაზს (შავ ზოლს) მიუყვება. ეს არის აქტიური ინფრაწითელი სენსორი, რომელიც ასხივებს და შემდეგ ახდენს თავისივე ინფრაწითელი სხივების აღმოჩენას. სენსორზე არსებული ინფრაწითელი შუქდიოდი მუდმივად ასხივებს ვიწროდ მიმართულ ინფრაწითელ სხივს, რომელიც დაბრკოლებაზე მოხვედრისას აირეკლება და მისი რაღაც ნაწილი ბრუნდება უკან. თუ ეს სხივები მოხვდება სენსორზე არსებულ ინფრაწითელ დეტექტორზე, მაშინ მოწყობილობის გამოსასვლელზე გვექნება დადებითი იმპულსი. თეთრი შუქდიოდი ასხივებს ინფრაწითელ სხივებს, ხოლო შავი - იღებს ინფრაწითელ სხივებს. ეს ინფრაწითელი სხივები უხილავია ჩვენი თვალისთვის (იხ. სურათი 1).



სურათი 1. ინფრაწითელი (IR) სენსორი და მისი მოქმედების პრინციპი

გასათვალისწინებელია, რომ შავი ზედაპირი თითქმის მთლიანად შთანთქავს შუქს და თუ შავ ზედაპირიან ობიექტს მივუშვერთ სენსორს, მაშინ გამოსხივებული ინფრაწითელი ტალღების მხოლოდ მცირე ნაწილი მოხვდება მიმღებზე და შეიძლება სენსორის დაფას არეკლილ სხივზე რეაქცია არც ჰქონდეს.

ამ სენსორებს აქვთ ასახვის მანძილი 1 მმ-დან 25 მმ-მდე და 30 გრადუსამდე. ჩვენ შეგვიძლია შევცვალოთ ეს მანძილი წინასწარ დაინსტალირებული კონტროლერით. ბაზარზე ამ სენსორების სხვადასხვა ტიპები არსებობს, მაგრამ ყველა ეს სენსორი ერთსა და იმავე ფუნქციას ასრულებს.

ინფრაწითელ (IR) სენსორს აქვს სამი საკონტაქტო პინი:

VCC - +5V

GND - დამიწება

Out - ციფრული მნიშვნელობა/სიდიდე

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino UNO
ENA	Pin 6
IN1	Pin 7
IN2	Pin 8
IN3	Pin 9
IN4	Pin 10
ENB	Pin 5

2. დააკავშირეთ ერთმანეთთან არდუინო და ინფრაწითელი (IR) სენსორები:

ინფრაწითელი (IR) სენსორი	Arduino UNO
ორივე სენსორის GND	GND
ორივე სენსორის 5V	5V
მარჯვენა სენსორის Out	Pin 11
მარცხენა სენსორის Out	Pin 12

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

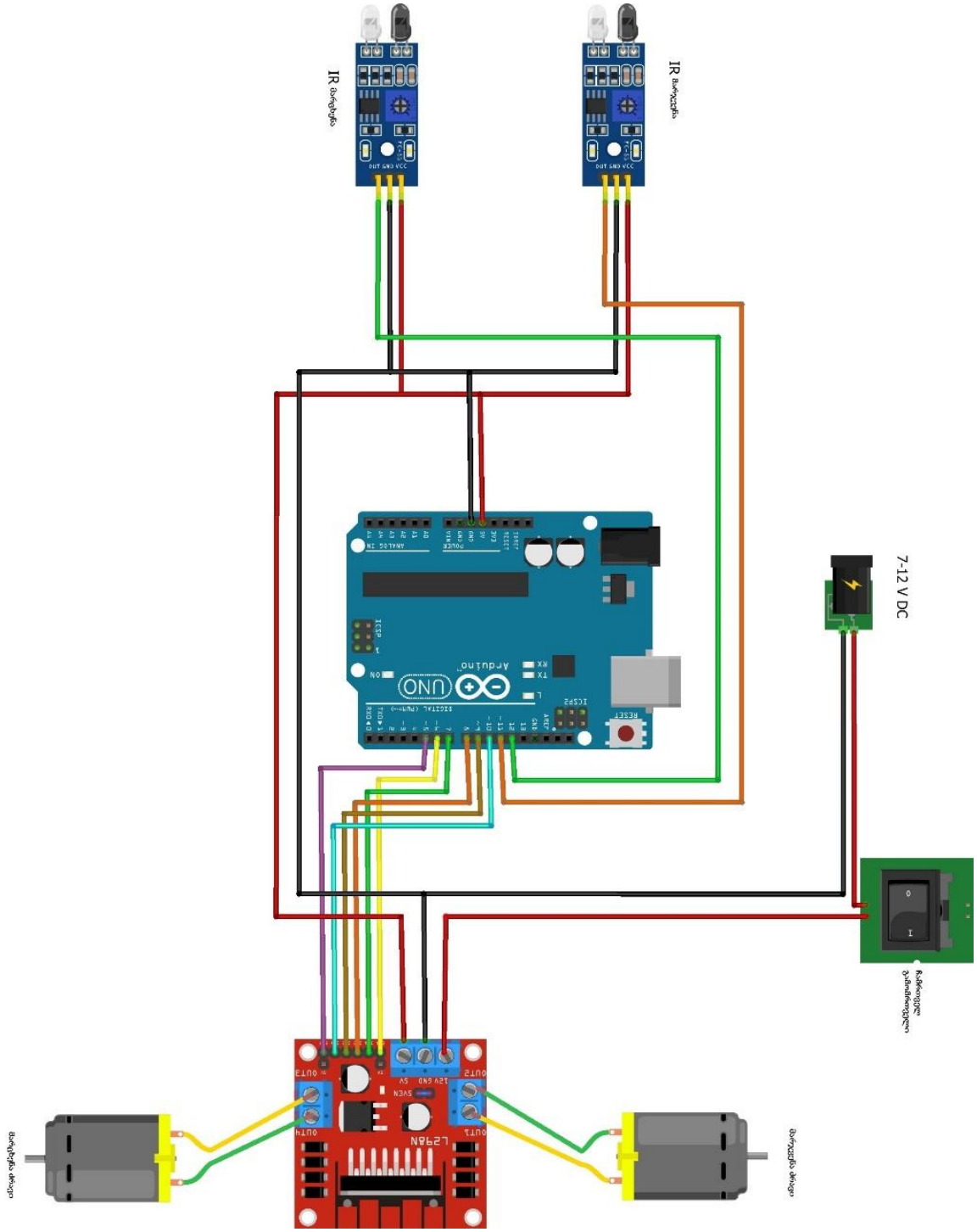
3. დააკავშირეთ არდუინო კომპიუტერთან.

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ პორტი.

4. ჩატვირთეთ კოდი.

გამორთეთ არდუინო კომპიუტერიდან. ჩაალაგეთ ელემენტები ბუდეში და „დაქოქეთ“ მანქანა.

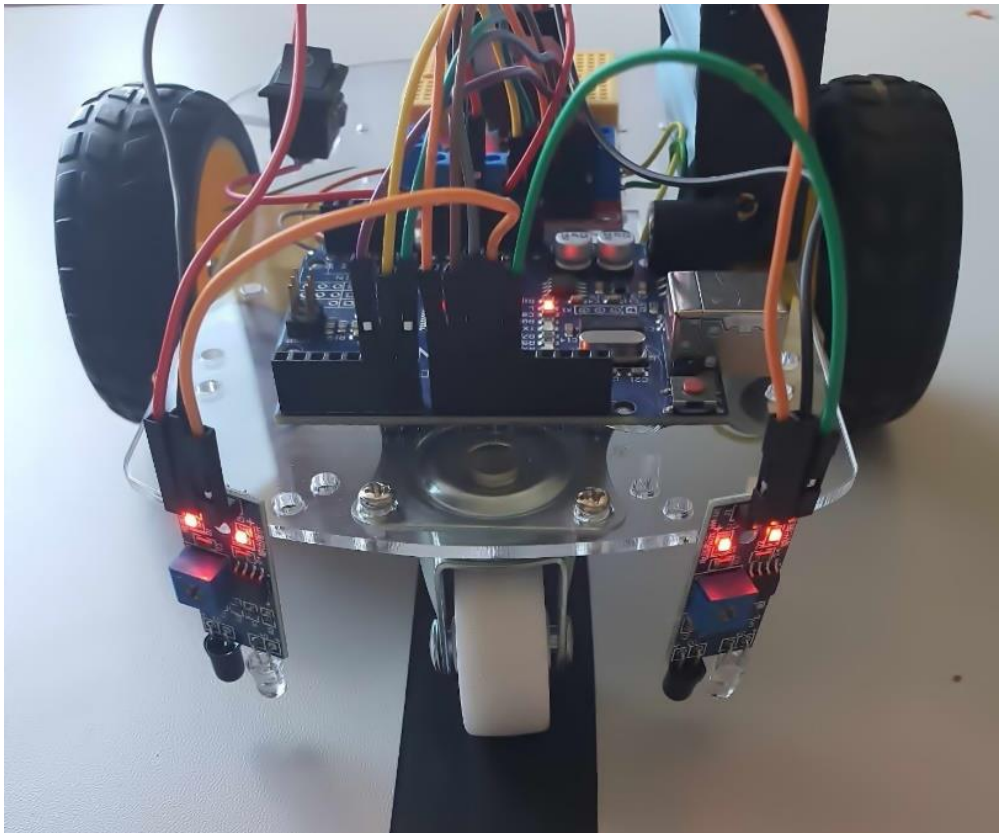


სურათი 2. სქემა

5. თქვენმა მანქანამ შავ ზოლზე რომ იმოძრაოს, საჭიროა ინფრაწითელი (IR) სენსორების დარეგულირება პოტენციომეტრით, რომელიც თავად სენსორზეა მოთავსებული (ლურჯი ღილაკი). იხილეთ სურათი 1.

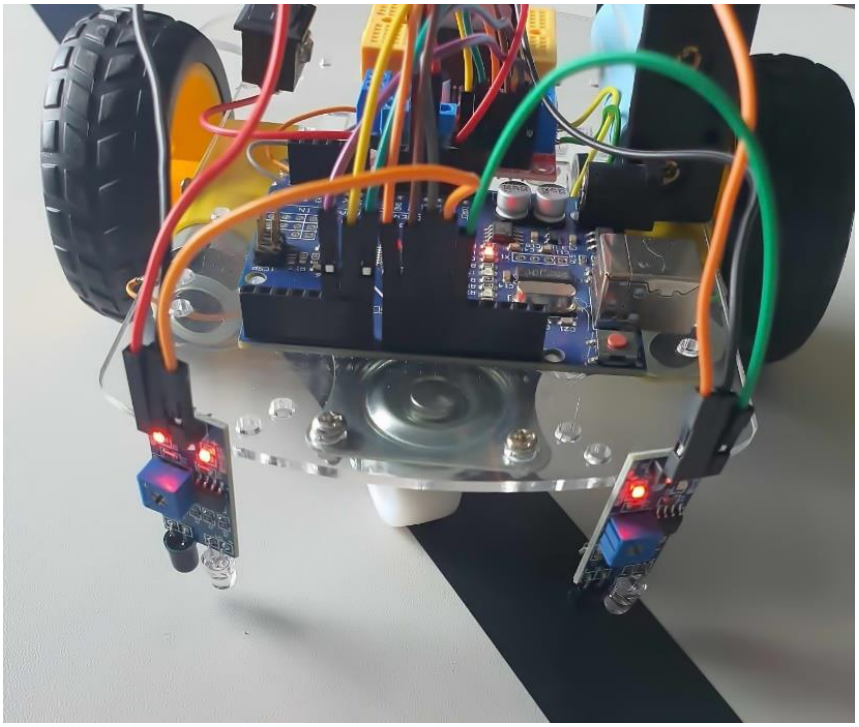
რა შემთხვევაში იმოძრავებს მანქანა შავ ზოლზე?

მანქანის „დაქოქვისას“ ორივე სენსორის პოტენციომეტრის თავზე ორი შუქდიოდი აინთება (იხ. სურათი 3).



სურათი 3. ინფრაწითელი (IR) სენსორების საწყისი პოზიცია

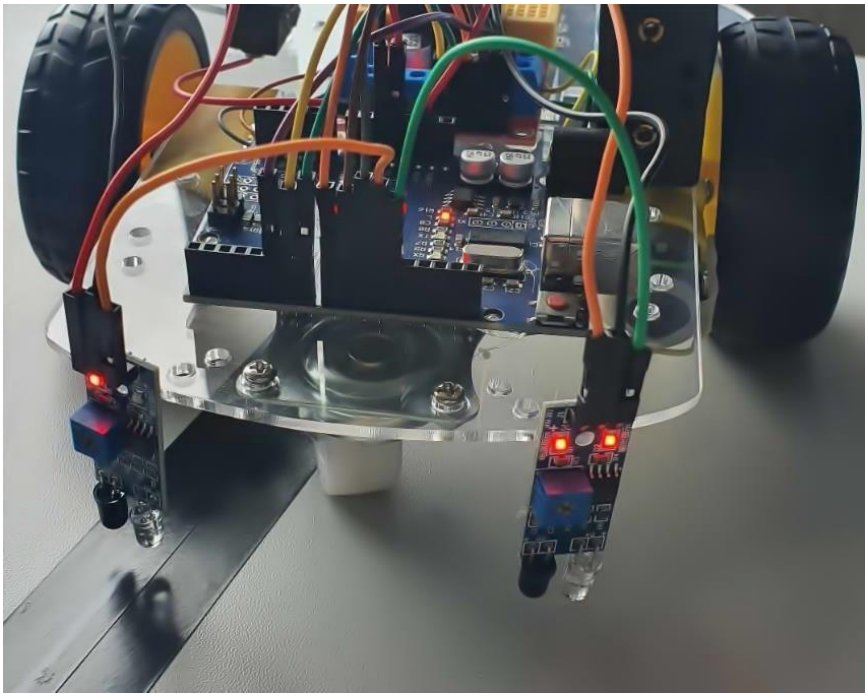
მანქანა დააფიქსირეთ ისე (ამ დროს მანქანა ხელში უნდა გეჭიროთ), რომ შავი ზოლი ორივე სენსორის შუაში მოხვდეს. ნელი მოძრაობით მიატრიალეთ მანქანა ისე, რომ ჯერ ერთი IR სენსორი მიაახლოვოთ შავ ზოლს და დააკვირდით: ორი მანათობელი შუქდიოდიდან ერთი უნდა ჩაქრეს (იხ. სურათი 4).



სურათი 4. მარჯვენა ინფრაწითელი (IR) სენსორი რეაგირებს შავ ფერზე

თუ ასე არ მოხდა, პოტენციომეტრი ისე უნდა დაარეგულიროთ, ვიდრე ამ შედეგს არ მიიღებთ.

იგივეს გაიმეორებთ მეორე სენსორის შემთხვევაშიც (იხ. სურათი 5).



სურათი 5. მარცხენა ინფრაწითელი (IR) სენსორი რეაგირებს შავ ფერზე

ახლა კი შეგიძლიათ ისიამოვნოთ თქვენი პროექტით.

კოდის მუშაობის პრინციპი

კოდი იყენებს ორ ინფრაწითელ (IR) სენსორს და ორ ძრავას რობოტის მოძრაობის კონტროლისთვის. გთავაზობთ კოდის მოკლე ახსნას:

1. სენსორები და ძრავები:

- IR სენსორები: გამოიყენება ორი IR სენსორი (IR_SENSOR_RIGHT და IR_SENSOR_LEFT), რომელიც ამოიცნობს შავ ზოლს.
- ძრავები: რობოტს აქვს ორი ძრავა (მარცხენა და მარჯვენა), რომლებიც კონტროლდება Arduino-ს პინების მეშვეობით.

2. კოდის ძირითადი ლოგიკა:

❖ **setup()**:

- ინიციალიზებულია სენსორების და ძრავების პინები.
- PWM სიხშირე შეცვლილია (TCCR0B = TCCR0B & B11111000 | B00000010), რათა ძრავებმა უფრო სწორად იმუშაონ დაბალი სიჩქარით.
- რობოტის ძრავები გაჩერებულია (rotateMotor(0, 0)).

❖ **loop()**:

- ყოველ ციკლში იკითხება IR სენსორების მნიშვნელობები (digitalRead(IR_SENSOR_RIGHT) და digitalRead(IR_SENSOR_LEFT)).
- რობოტი მოძრაობს შემდეგი ლოგიკით:
  - თუ ორივე სენსორი არ ამოიცნობს შავ ხაზს (LOW), რობოტი წინ მიიწევს (rotateMotor(MOTOR_SPEED, MOTOR_SPEED)).
  - თუ მარჯვენა სენსორი ამოიცნობს შავ ხაზს (HIGH), რობოტი მარჯვნივ მობრუნდება (rotateMotor(-MOTOR_SPEED, MOTOR_SPEED)).
  - თუ მარცხენა სენსორი ამოიცნობს შავ ხაზს (HIGH), რობოტი მარცხნივ მობრუნდება (rotateMotor(MOTOR_SPEED, -MOTOR_SPEED)).
  - თუ ორივე სენსორი ამოიცნობს შავ ხაზს, რობოტი გაჩერდება (rotateMotor(0, 0)).

3. ძრავების კონტროლი:

- rotateMotor(int rightMotorSpeed, int leftMotorSpeed): ეს ფუნქცია აკონტროლებს ძრავების მიმართულებას და სიჩქარეს.
    - თუ სიჩქარე დადებითია (> 0), ძრავა მოძრაობს წინ.
    - თუ სიჩქარე უარყოფითია (< 0), ძრავა მოძრაობს უკან.
    - თუ სიჩქარე ნულის ტოლია (0), ძრავა გაჩერებულია.
    - analogWrite() გამოიყენება ძრავების სიჩქარის კონტროლისთვის.

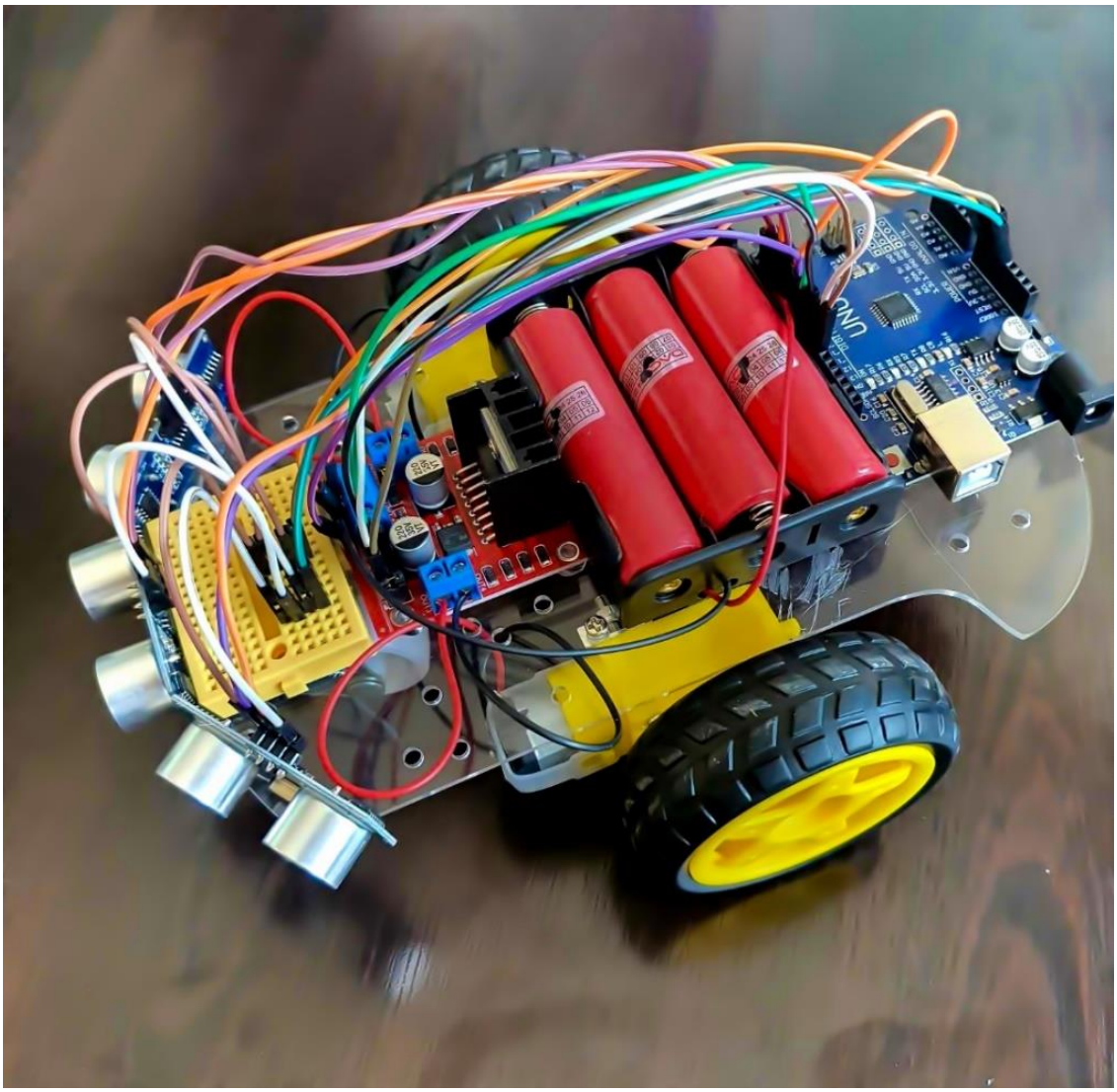
4. პარამეტრები:

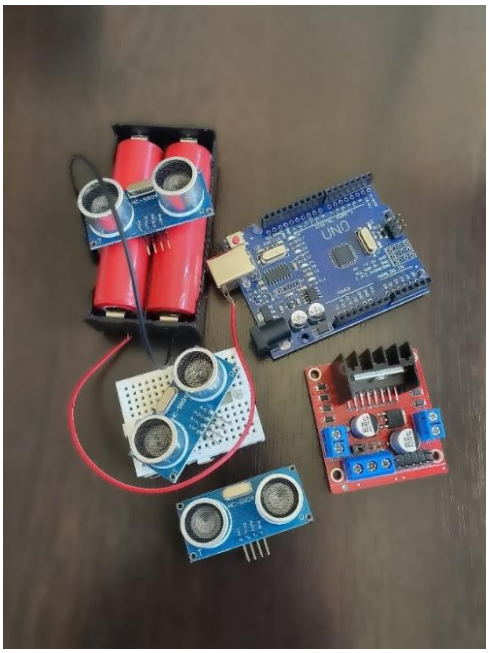
- MOTOR_SPEED: ძრავების სიჩქარე (180).
- IR_SENSOR_RIGHT და IR_SENSOR_LEFT: IR სენსორების პინები (11 და 12).

5. PWM სიხშირის შეცვლა:

TCCR0B = TCCR0B & B11111000 | B00000010: ეს ხაზი ცვლის PWM სიხშირეს D5 და D6 პინებზე, რათა ძრავებმა უფრო სწორად იმუშაონ დაბალი სიჩქარით.

ობიექტს ადევნებული რობოტი მანქანა





საჭირო რესურსი

- Arduino UNO
- 2WD რობოტი მანქანა (კომპლექტი)
- პატარა სამაკეტო დაფა
- გამტარები: დედალ-მამალი, მამალ-მამალი
- ძრავას დრაივერი L298N
- 3 ცალი ულტრაბგერითი სენსორი
- 2 ცალი ლითიუმის ელემენტი 3.7V და ბუდე

საჭირო ბიბლიოთეკა

- New Ping

როგორ მუშაობს პროექტი

ბოლო წლებში რობოტიკამ მნიშვნელოვანი წინსვლა განიცადა, რამაც შესაძლებელი გახადა ინტელექტუალური მანქანების შექმნა, რომლებსაც შეუძლიათ გარემოსთან ურთიერთქმედება. რობოტიკის ერთ-ერთი საინტერესო მიმართულებაა მანქანა, რომელიც ადამიანს (მოძრავ ობიექტს) აედევნება. ამ რობოტებს შეუძლიათ ადამიანის ავტონომიურად მიკვლევა და მიყოლა, რაც მათ სასარგებლოს ხდის სხვადასხვა შემთხვევებში, მაგალითად, როგორიცაა დახმარება ხალხმრავალ ადგილებში, ნავიგაციის მხარდაჭერა ან, თუნდაც, როგორც ადამიანის თანამგზავრი (კომპანიონი). ამ პროექტში ჩვენ დეტალურად განვიხილავთ, თუ როგორ შევქმნათ ადამიანს ან ნებისმიერ მოძრავ ობიექტს ადევნებული რობოტი Arduino UNO-სა და სამი ულტრაბგერითი სენსორის გამოყენებით.

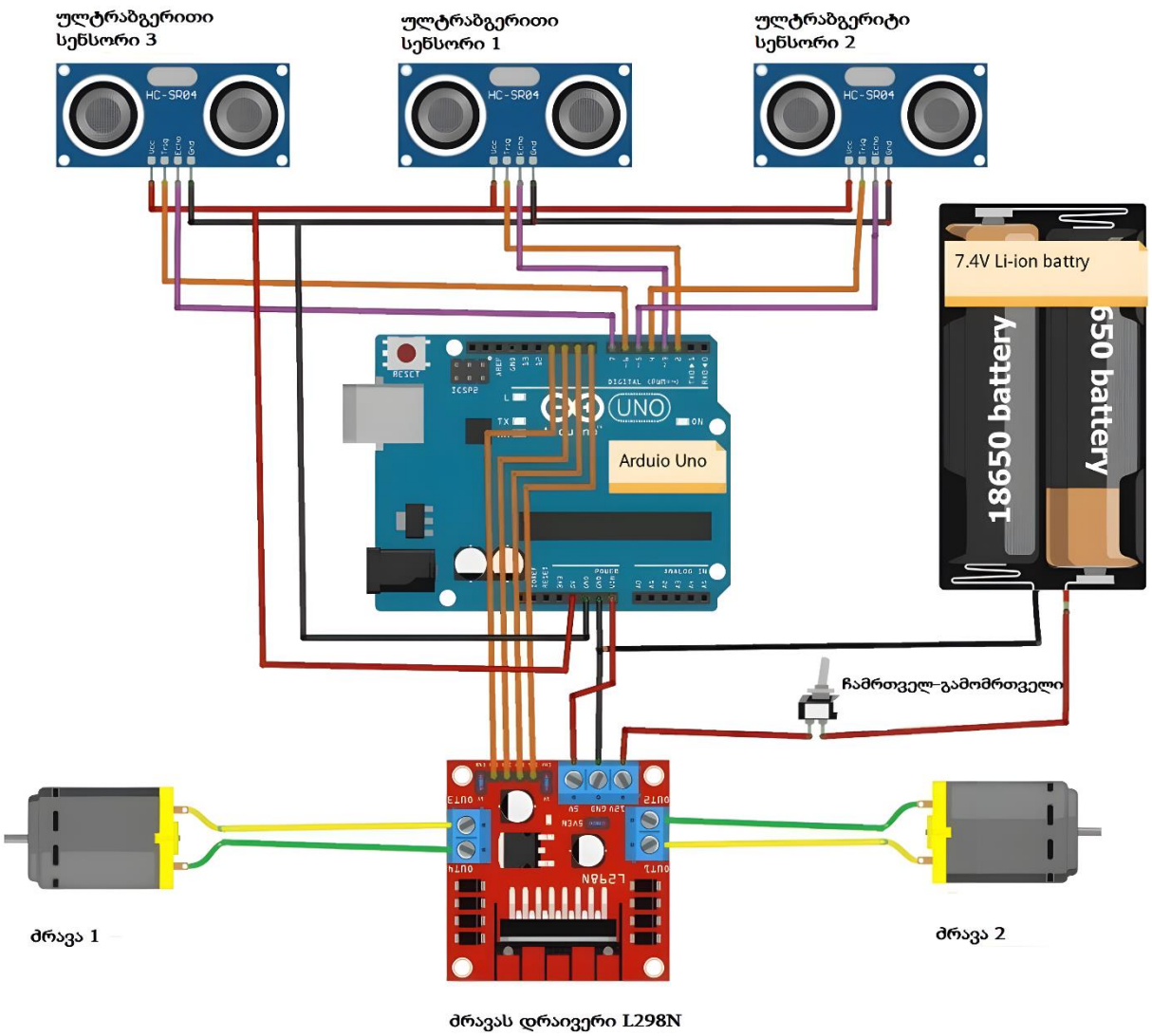
ასეთი რობოტის შექმნა Arduino-ს და სამი ულტრაბგერითი სენსორის გამოყენებით საინტერესო პროექტია. რაც ამ პროექტს განსაკუთრებულად საინტერესოს ხდის, არის ის, რომ მასში გამოყენებულია არა მხოლოდ ერთი, არამედ სამი ულტრაბგერითი სენსორი (ულტრაბგერითი სენსორის მუშაობის პრინციპი და მახასიათებლები იხილეთ I პროექტში). ეს ახალ განზომილებას მატებს გამოცდილებას, რადგან, როგორც წესი, ასეთი რობოტები შექმნილია ერთი ულტრაბგერითი, ორი ინფრაწითელი (IR) სენსორითა და ერთი Servo ძრავის გამოყენებით. ამ Servo ძრავას არ აქვს რაიმე განსაკუთრებული როლი პროცესში. ულტრაბგერითი სენსორების საშუალებით შეგიძლიათ გაზომოთ მანძილი და გამოიყენოთ ეს ინფორმაცია ნავიგაციისთვის და ადამიანის (ან ნებისმიერი მოძრავი ობიექტის) ასადევნებლად. ქვემოთ მოცემულია სქემა (იხ. სურათი 1), რომელიც საჭიროა ასეთი რობოტის შესაქმნელად.

სქემა მოიცავს სამ ულტრაბგერით სენსორს, რომელიც საშუალებას იძლევა გავზომოთ მანძილი სამი მიმართულებით: წინ, მარჯვნივ და მარცხნივ. ეს სენსორები უკავშირდება Arduino-ს შესაბამისი ციფრული კონტაქტების საშუალებით. გარდა ამისა, სქემა მოიცავს ორ DC ძრავას მოძრაობისთვის, რომელიც დაკავშირებულია L298N ძრავას დრაივერის მოდულთან, რომელიც, თავის მხრივ, უკავშირდება არდუინოს შესაბამისი ციფრული კონტაქტების გამოყენებით. ორი 3.7V ლითიუმის ელემენტი კი გამოიყენება კვების წყაროდ, რომელიც დაკავშირებულია ძრავას დრაივერის მოდულთან ჩამრთველ-გამომრთველის საშუალებით.

მთლიანობაში, ეს დიაგრამა გვიჩვენებს მოძრავ ობიექტს ადევნებული რობოტის ეფექტიანი მუშაობისთვის საჭირო ძირითად კომპონენტებსა და კავშირებს.

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).



სურათი 1. სქემა

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრავერი:

ძრავას დრაივერი L298N	Arduino UNO
IN1	Pin 8
IN2	Pin 9
IN3	Pin 10
IN4	Pin 11

2. დააკავშირეთ ერთმანეთთან ულტრაბგერითი სენსორები და არდუინო:

ულტრაბგერითი სენსორი 3	Arduino UNO
VCC (5V)	5V
Trig	Pin 6
Echo	Pin 7
GND	GND

ულტრაბგერითი სენსორი 1	Arduino UNO
VCC (5V)	5V
Trig	Pin 2
Echo	Pin 3
GND	GND

ულტრაბგერითი სენსორი 2	Arduino UNO
VCC (5V)	5V
Trig	Pin 4
Echo	Pin 5
GND	GND

3. დააკავშირეთ ერთმანეთთან არდუინო, ძრავას დრაივერი და ულტრაბგერითი სენსორები სამაკეტო დაფაზე

Arduino UNO	სამაკეტო დაფა
5V	+
GND	GND

ძრავას დრაივერი L298N	სამაკეტო დაფა
+5V	+
GND	GND

ულტრაბგერითი სენსორი (3)	სამაკეტო დაფა
VCC	+
GND	GND

4. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ პორტი.

5. ჩატვირთეთ კოდი.

6. გამორთეთ არდუინო კომპიუტერიდან. ჩაალაგეთ ელემენტები ბუდეში და „დაქოქეთ“ მანქანა.

კოდის მუშაობის პრინციპი

კოდი მუშაობს Arduino-სა და სამი ულტრაბგერითი სენსორის გამოყენებით, რათა რობოტმა შეძლოს ადამიანის მოძებნა და მიყოლა.

1. **მანძილის გაზომვა**: სამი ულტრაბგერითი სენსორი განლაგებულია რობოტის წინ, მარცხნივ და მარჯვნივ. ისინი გაზომავენ მანძილს მოძრავ ობიექტამდე.
2. **მონაცემთა ანალიზი**: Arduino იღებს მონაცემებს სენსორებიდან და აფასებს, სად არის ობიექტი (წინ, მარცხნივ თუ მარჯვნივ).
3. **მიყოლის ლოგიკა**:
    - თუ ობიექტი წინა სენსორის მიმართულებით არის, რობოტი წინ მიიწევს.
    - თუ ობიექტი მარცხნივ ან მარჯვნივ არის, რობოტი შესაბამისად მობრუნდება.
    - თუ ადამიანი ახლოს არ არის, რობოტი გაჩერდება ან დაიწყებს ძიებას.
4. **მოძრაობის კონტროლი**: Arduino აკონტროლებს ძრავებს, რათა რობოტმა შეძლოს მოძრაობა და მიყოლა.

ასეთია ამ პროექტის კოდის ძირითადი ნაბიჯები. უფრო დაწვრილებით კი:

1. სენსორები და ძრავები:

- **სენსორები:** გამოიყენება სამი ულტრაბგერითი სენსორი (S1, S2, S3), რომელიც ზომავს მანძილს წინ, მარცხნივ და მარჯვნივ.
- **ძრავები:** რობოტს აქვს ორი ძრავა (მარცხენა და მარჯვენა), რომელიც კონტროლდება Arduino-ს პინების მეშვეობით.

2. კოდის ძირითადი ლოგიკა:

- **setup():** აქ ინიციალიზებულია სენსორების და ძრავების პინები, ასევე მიმდევრობითი კომუნიკაცია.
- **loop():**
    - ყოველ ციკლში იზომება მანძილი სამივე სენსორით (sensorOne(), sensorTwo(), sensorThree()).

- კოდი ადგენს, რომელი სენსორი არის ყველაზე ახლოს ობიექტთან (ან დაბრკოლებასთან).
- რობოტი მოძრაობს შემდეგი ლოგიკით:
  - თუ წინა სენსორი აღმოაჩენს ობიექტს (frontDistance < MAX_DISTANCE), რობოტი წინ მიიწევს.
  - თუ მარცხენა სენსორი აღმოაჩენს ობიექტს (leftDistance < rightDistance), რობოტი მარცხნივ მობრუნდება.
  - თუ მარჯვენა სენსორი აღმოაჩენს ობიექტს, რობოტი მარჯვნივ მობრუნდება.
  - თუ მოძრავი ობიექტი ძალიან ახლოს არის (frontDistance < MIN_DISTANCE_BACK), რობოტი უკან დაიხევს.
  - თუ არავინ არ არის აღმოჩენილი, რობოტი გაჩერდება.

3. სენსორების ფუნქციები:

- sensorOne(), sensorTwo(), sensorThree(): ეს ფუნქციები იყენებენ ულტრაბგერით სენსორებს მანძილის გასაზომად. ისინი აგზავნიან იმპულსს (Trig) და იღებენ დაბრუნებულ იმპულსს (Echo), რომელიც გარდაიქმნება მანძილად (სანტიმეტრებში).

4. ძრავების კონტროლი:

- moveForward(): ძრავები მოძრაობენ წინ.
- moveBackward(): ძრავები მოძრაობენ უკან.
- turnLeft(): რობოტი მარცხნივ მობრუნდება.
- turnRight(): რობოტი მარჯვნივ მობრუნდება.
- stop(): ძრავები გაჩერებულია.

5. პარამეტრები:

- MAX_DISTANCE: მაქსიმალური მანძილი, რომლის დროსაც რობოტი რეაგირებს (40 სმ).
- MIN_DISTANCE_BACK: მინიმალური მანძილი, რომლის დროსაც რობოტი უკან დაიხევს (5 სმ).
- MAX_SPEED და MIN_SPEED: ძრავების სიჩქარის ლიმიტები.

6. გამართვა:

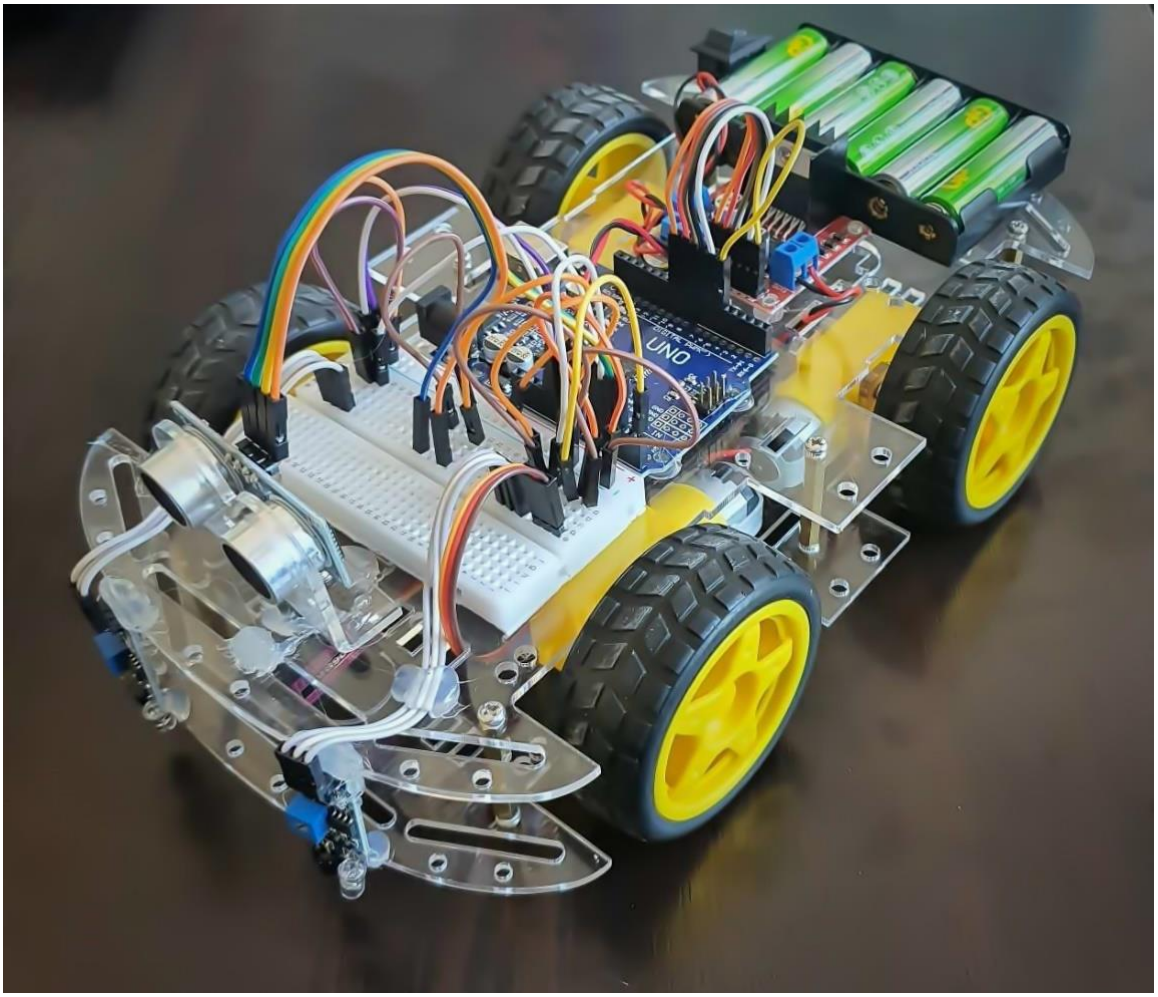
- მიმდევრობითი კომუნიკაციის მონიტორი (Serial.print()) გამოიყენება მანძილების და რობოტის მოქმედებების საჩვენებლად, რაც გამართვას ამარტივებს.

7. დაყოვნება:

- delay(100): ყოველ ციკლში გამოყენებულია დაყოვნება, რათა თავიდან იქნას აცილებული ზედმეტი მონაცემების წაკითხვა. ასევე, სტაბილურობის გასაუმჯობესებლად.

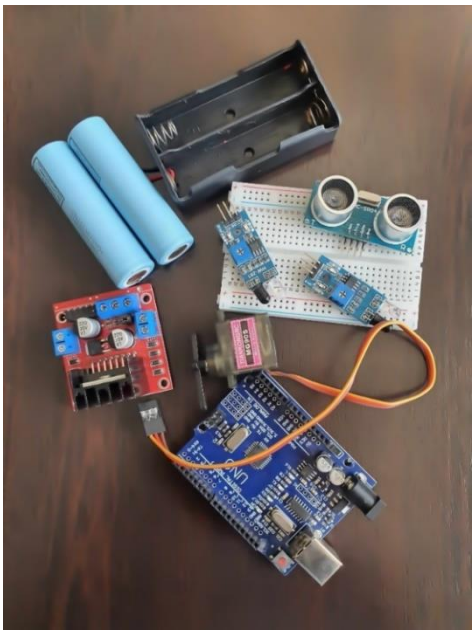
IV პროექტი

ხაზზე მოძრაობა და დაბრკოლებების გვერდის ავლა



საჭირო რესურსი

- Arduino UNO
- 4WD ჭკვიანი მანქანა (კომპლექტი)
- სამაკეტო დაფა
- გამტარები: მამალ-მამალი, დედალ-მამალი
- ძრავას დრაივერი L298N
- ულტრაბგერითი სენსორი
- 2 ცალი ინფრაწითელი (IR) სენსორი
- Servo ძრავა Tower Pro SG90 9g
- 2 ცალი ლითიუმის ელემენტი 3.7V და ბუდე



საჭირო ბიბლიოთეკა

- New Ping

როგორ მუშაობს პროექტი

ამ პროექტში ჩვენ გამოვიყენებთ არდუინოს და L298N ძრავას დრაივერს, რათა შევქმნათ ზოლზე მოძრავი რობოტი, რომელსაც შეუძლია თავიდან აიცილოს დაბრკოლებები. ამით ჩვენ ერთ რობოტში ვაერთიანებთ I და II პროექტს და ერთი მანქანით ვქმნით მოქნილ და დინამიურ რობოტს, რომელიც თავის თავში აერთიანებს ზოლზე მოძრაობასა და დაბრკოლებების გვერდის ავლას. არდუინოსა და სხვადასხვა კომპონენტების დახმარებით თქვენ შექმნით ინტელექტუალურ რობოტს, რომელსაც შეუძლია რთულ პირობებში ნავიგაცია. ქვემოთ მოცემულია სქემა (იხ. სურათი 1), რომელიც საჭიროა ასეთი რობოტის შესაქმნელად.

პროექტში გამოყენებულ რესურსზე აქ აღარ შევჩერდებით, რადგან მათზე დაწვრილებით ვისაუბრეთ წინა პროექტებში.

ავაწყოთ სქემა

სქემა მოიცავს ულტრაბგერით სენსორს, რომელიც უკავშირდება Arduino-ს შესაბამისი ციფრული კონტაქტების საშუალებით. გარდა ამისა, სქემა მოიცავს ოთხ DC ძრავას მოძრაობისთვის, რომლებიც დაკავშირებულია L298N ძრავას დრაივერის მოდულთან, რომელიც, თავის მხრივ, უკავშირდება არდუინოს შესაბამისი ციფრული კონტაქტების გამოყენებით. 2 ცალი 3.7V ლითიუმის ელემენტი კი გამოიყენება კვების წყაროდ, რომელიც დაკავშირებულია ძრავის დრაივერის მოდულთან ჩამრთველ-გამომრთველის საშუალებით (იხ. სურათი 1).

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino UNO
ENA	Pin 10
IN1	Pin 9
IN2	Pin 8
IN3	Pin 7
IN4	Pin 6
ENB	Pin 5

2. დააკავშირეთ ერთმანეთთან არდუინო და ინფრაწითელი სენსორები:

ინფრაწითელი (IR) სენსორი		Arduino UNO
ორივე სენსორის	GND	GND
ორივე სენსორის	5V	5V
მარჯვენა სენსორის	Out	Pin A1
მარცხენა სენსორის	Out	Pin A0

3. დააკავშირეთ ერთმანეთთან არდუინო და Srevo ძრავა:

Servo ძრავა	Arduino UNO
შავი გამტარი (GND)	GND
წითელი გამტარი (5V)	5V
ყვითელი გამტარი	Pin A5

4. დააკავშირეთ ერთმანეთთან ულტრაბგერითი სენსორი და არდუინო:

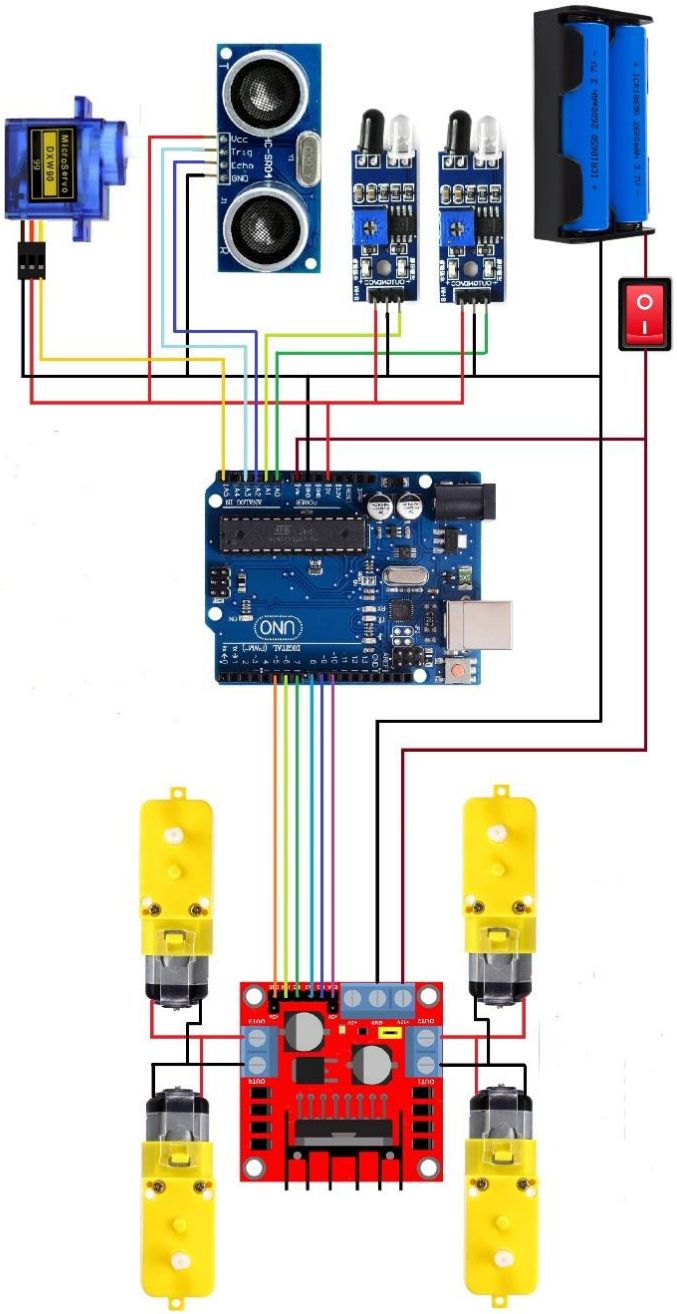
ულტრაბგერითი სენსორი	Arduino UNO
VCC (5V)	5V
Trig	Pin A3
Echo	Pin A2
GND	GND

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

5. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი, Servo ძრავა, ულტრაბგერითი სენსორი და ინფრაწითელი სენსორები (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino UNO	სამაკ. დაფა	ძრავას დრაივერი L298N	სამაკ. დაფა	Servo ძრავა	სამაკ. დაფა	ულტრაბგერითი სენსორი	სამაკ. დაფა	IR სენსორები	სამაკ. დაფა
5V	+	+5V	+	წით. გამტარი	+	VCC	+	VCC	+
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND



სურათი 1. სქემა

6. დააკავშირეთ არდუინო კომპიუტერთან.

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ პორტი.

7. ჩატვირთეთ კოდი.

გამორთეთ არდუინო კომპიუტერიდან. ჩაალაგეთ ელემენტები ბუდეში და „დაქოქეთ“ მანქანა.

კოდის მუშაობის პრინციპი

ეს კოდი აკონტროლებს რობოტს, რომელსაც აქვს:

- IR სენსორები (მარცხენა და მარჯვენა) — ხაზის მიმდევრობისთვის.
- ულტრაბგერითი სენსორი — დაბრკოლებების აღმოსაჩენად.
- Servo მოტორი — ულტრაბგერითი სენსორის მიმართულების შესაცვლელად.
- L298 ძრავას დრაივერი — ოთხი DC ძრავას კონტროლისთვის.

❖ **ინიციალიზაცია (setup()):**

- პინები კონფიგურირებულია, როგორც შესაბამისი შემავალ/გამომავალი.
- IR სენსორები (L_S და R_S) კონფიგურირებულია, როგორც შემავალი.
- ულტრაბგერითი სენსორი (echo და trigger) კონფიგურირებულია.
- მოტორები (in1, in2, in3, in4, enA, enB) კონფიგურირებულია, როგორც გამომავალი.
- Servo ძრავა ინიციალიზებულია და ატრიალებს ულტრაბგერით სენსორს 70°-დან 140°-მდე და უკან.

❖ **მთავარი ციკლი (loop()):**

- ყოველ ჯერზე კოდი ამოწმებს:
  - **IR სენსორების მდგომარეობას** (ხაზის მიმდევრობა).
  - **ულტრაბგერითი სენსორის მონაცემებს** (დაბრკოლებების აღმოჩენა).

❖ **ხაზის მიმდევრობა:**

- თუ ორივე IR სენსორი (L_S და R_S) ამოიცნობს თეთრს (ხაზი არ არის), რობოტი მოძრაობს წინ (forword()), თუ წინ არ არის დაბრკოლება.
- თუ მარჯვენა IR სენსორი (R_S) ამოიცნობს შავს (ხაზი), რობოტი უხვევს მარჯვნივ (turnRight()).
- თუ მარცხენა IR სენსორი (L_S) ამოიცნობს შავს, რობოტი უხვევს მარცხნივ (turnLeft()).

❖ **დაბრკოლებები:**

- **თუ ულტრაბგერითი სენსორი** ამოიცნობს დაბრკოლებას (distance_F <= Set), რობოტი:
  - ჩერდება (Stop()).
  - ატრიალებს Servo ძრავას, რომ გაზომოს მანძილი მარცხნივ (distance_L) და მარჯვნივ (distance_R).
  - ადარებს მანძილებს (compareDistance()) და ირჩევს უსაფრთხო მიმართულებას.
  - მოძრაობს არჩეული მიმართულებით.

❖ **მოტორების კონტროლი:**

- forword(): ორივე ძრავა მოძრაობს წინ.
- backword(): ორივე ძრავა მოძრაობს უკან.
- turnRight(): მარცხენა ძრავა მოძრაობს წინ, მარჯვენა - უკან.
- turnLeft(): მარჯვენა ძრავა მოძრაობს წინ, მარცხენა - უკან.
- Stop(): ორივე ძრავა ჩერდება.

❖ **Servo ძრავას კონტროლი:**

- servoPulse(): ატრიალებს Servo ძრავას კონკრეტულ კუთხეზე, რათა ულტრაბგერითი სენსორით გაზომოს მანძილი სხვადასხვა მიმართულებით.

❖ **ულტრაბგერითი სენსორის მუშაობა:**

- Ultrasonic_read(): გაგზავნის ბგერის იმპულსს (trigger) და ზომავს დროს, რომელიც სჭირდება ექოს უკან დასაბრუნებლად (echo). ეს დრო გარდაიქმნება მანძილად.

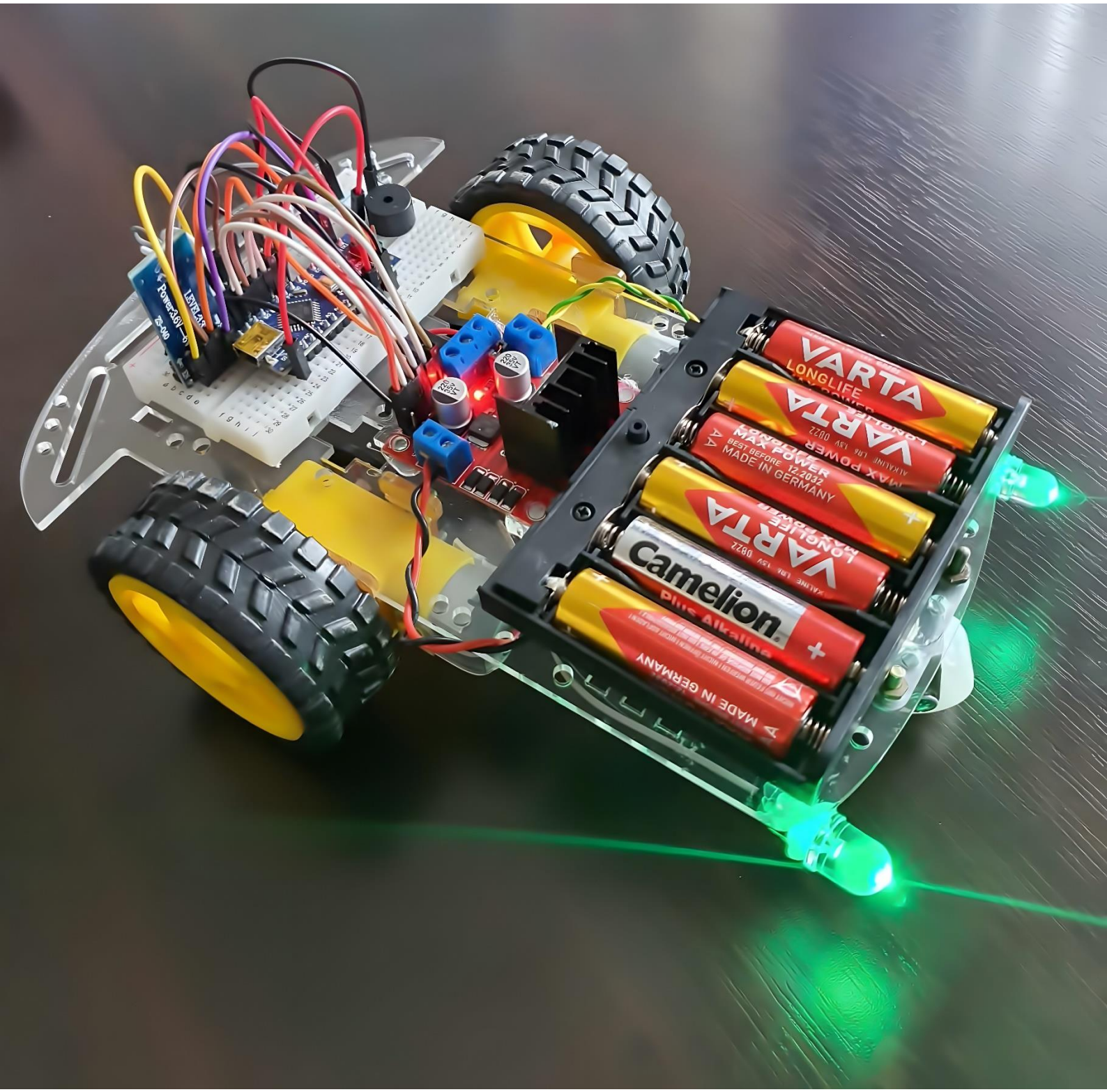
კოდის ძირითადი ლოგიკა:

1. **თუ არ არის დაბრკოლება:**
   - რობოტი მიჰყვება ხაზს IR სენსორების მიხედვით.
2. **თუ არის დაბრკოლება:**
   - რობოტი ჩერდება.
   - ამოწმებს მარცხნივ და მარჯვნივ მანძილებს.
   - ირჩევს უსაფრთხო მიმართულებას და მოძრაობს.

სცენარი:

1. რობოტი მოძრაობს წინ და მიჰყვება ხაზს.
2. წინ გამოჩნდება დაბრკოლება.
3. რობოტი ჩერდება და ამოწმებს მარცხნივ და მარჯვნივ.
4. თუ მარცხნივ მეტი ადგილია, უხვევს მარცხნივ და აგრძელებს მოძრაობას.
5. თუ მარჯვნივ მეტი ადგილია, უხვევს მარჯვნივ და აგრძელებს მოძრაობას.

რობოტი მანქანის მართვა ბლუთუზით

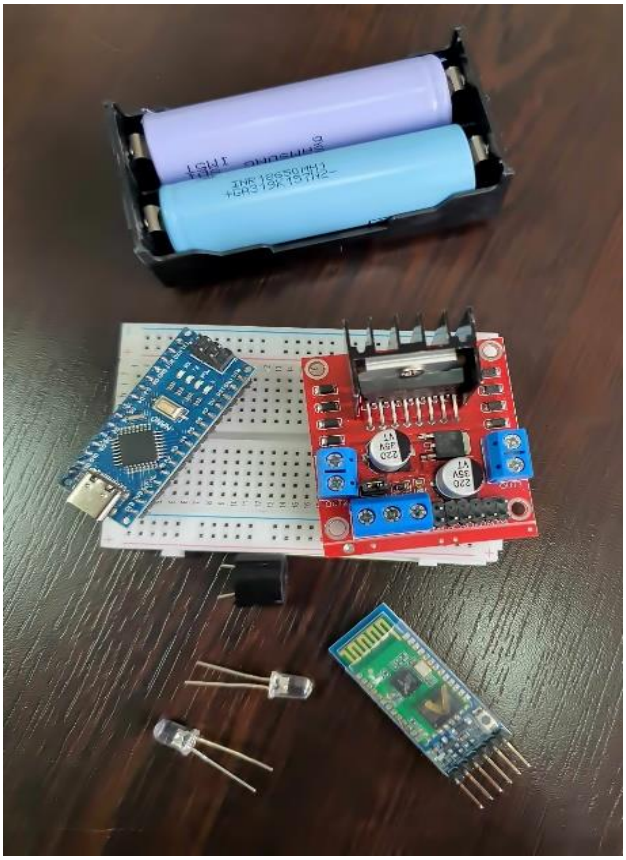


საჭირო ბიბლიოთეკა

- Tone

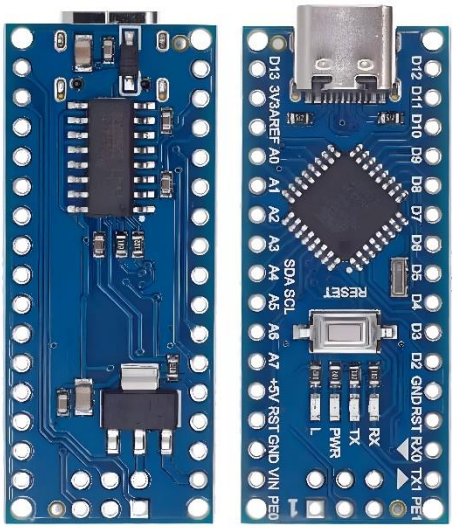
საჭირო რესურსი

- Arduino Nano
- 2WD რობოტი მანქანა (კომპლექტი)
- სამაკეტო დაფა
- გამტარები: მამალ-მამალი და დედალ-მამალი
- 2 ცალი LED ნათურა
- პიეზო დინამიკი
- Bluetooth HC-05 მოდული
- ძრავას დრაივერი L298N
- 6 ცალი 1.5 V ელემენტი და ბუდე (შეგიძლიათ გამოიყენოთ 2 ცალი ლითიუმის ელემენტი + ბუდე)
- Bluetooth კონტროლერი SriTu Hobby



როგორ მუშაობს პროექტი

ამ პროექტში ჩვენ შევქმნით 2WD მანქანას, რომელიც იმართება ბლუთუზით. სქემა იხილეთ ქვემოთ (სურათი 3). კვლავ გამოვიყენებთ Arduino Nano-ს. Arduino Nano შედარებით მცირე ზომებით განსხვავდება Arduino Uno-სგან. შესაბამისად, დაფაც უფრო მცირე ზომისაა (იხ. სურათი 1).



სურათი 1. Arduino Nano

რა განსხვავებაა Arduino Uno-სა და Arduino Nano-ს შორის

ქვემოთ მოცემული ცხრილი მოკლედ აჯამებს მთავარ პარამეტრებს:

მახასიათებელი	Arduino Uno	Arduino Nano
ძირითადი განსხვავება	**სტანდარტული ფორმ-ფაქტორი**	**კომპაქტურობა**
მიკროკონტროლერი	ATmega328P	ATmega328P (იგივე!)
სამუშაო ძაბვა	5V	5V
დიგიტალური I/O პინები	14	14 (იგივე)
ანალოგური პინები	6	8 (2-ით მეტი)
ფლეშ მეხსიერება	32 KB	32 KB (იგივე)
SRAM	2 KB	2 KB (იგივე)
EEPROM	1 KB	1 KB (იგივე)
ზომა	დიდი (~68.6 x 53.4 მმ)	**ძალიან პატარა** (~18 x 45 მმ)
USB კავშირი	დედაპლატაზე მოწყობილი USB-B პორტი	**Mini-USB** ან **Micro-USB** (მოდელის მიხედვით)
დენის მიწოდება	დედაპლატაზე მოწყობილი Jack კონექტორი	არ აქვს ცალკე Jack, მხოლოდ Vin პინი ან USB-დან
ფასი	ჩვეულებრივ, უფრო ძვირი	**ჩვეულებრივ, უფრო იაფი** (განსაკუთრებით, კლონები)

მნიშვნელოვანი პრაქტიკული განსხვავებები, რომლებიც გავლენას ახდენენ არჩევანზე:

1. ფორმ-ფაქტორი და დაკავშირება

Uno: გააჩნია **სტანდარტული პინები**, რაც იმას ნიშნავს, რომ შეგიძლიათ პირდაპირ დაამაგროთ **სამაკეტო დაფაზე** (თუმცა ის დიდ ადგილს იკავებს) ან გამოიყენოთ მისთვის განკუთვნილი Shield-ები (დამატებითი დაფები, რომლებიც ზემოდან ერგება). ეს არის მისი უდიდესი უპირატესობა.

- **Nano:** შექმნილია იმისთვის, რომ **დამაგრდეს სამაკეტო დაფაზე**. ის იკავებს მინიმალურ ადგილს და იდეალურია კომპაქტური პროექტებისთვის.

2. USB ინტერფეისი

- **Uno:** იყენებს დიდ **USB-B** კაბელს (როგორც ძველ პრინტერებში). ეს კაბელი ადვილად მოიძებნება და მტკიცეა.
- **Nano:** იყენებს **Mini-USB** ან **Micro-USB** კაბელს (უფრო ახალ მოდელებში). ეს კაბელები უფრო თანამედროვეა.

3. დენის მიწოდება

- **Uno:** აქვს ცალკე **Jack კონექტორი**. ეს ძალიან მოსახერხებელია, როდესაც პროექტს USB-ს გაუთიშავთ და გინდათ გარე წყაროდან (მაგ., battery pack) მიაწოდოთ დენი.
- **Nano: არ აქვს** ცალკე Jack. დენის მიწოდება შეგიძლიათ მხოლოდ USB-ს ან **Vin პინის** მეშვეობით, რაც ზოგჯერ ნაკლებად მოსახერხებელია.

4. ფასი და ხელმისაწვდომობა

Nano (განსაკუთრებით არაოფიციალური კლონები, როგორიცაა CH340 ჩიპიანი) **ჩვეულებრივ, გაცილებით იაფია,** ვიდრე Uno. ეს მას პოპულარულს ხდის დამწყებთათვის და იმ პროექტებისთვის, სადაც ბევრი კომპონენტის გამოყენება გჭირდებათ.

დასკვნა:

აირჩიოთ Arduino Uno:

- **დამწყები ხართ:** მისი სტანდარტული ზომა და მოწყობა უფრო ადვილია სამუშაოდ.
- **გჭირდებათ Shield-ების გამოყენება:** მრავალი მოსახერხებელი დამატებითი დაფა (Ethernet, Motor Driver, SD Card) შექმნილია კონკრეტულად Uno-ს ფორმ-ფაქტორისთვის.
- **გჭირდებათ მოხერხებული გარე დენის მიწოდება** (ბატარეიდან).

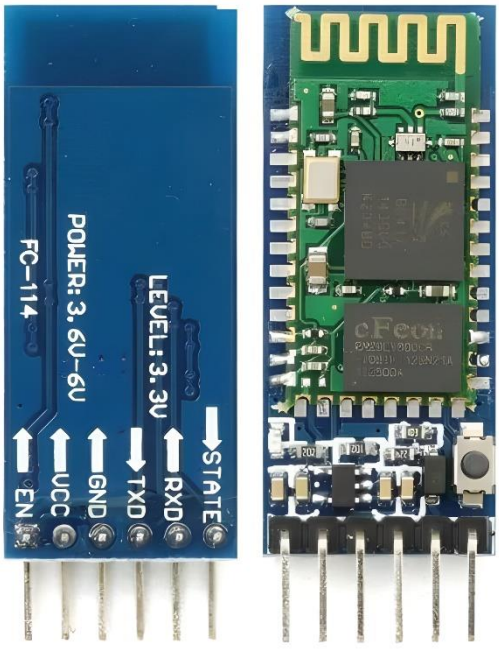
თუ, აირჩიოთ Arduino Nano:

- **ქმნით კომპაქტურ პროექტს** (მაგ., რობოტი, პორტატული მოწყობილობა).
- **გჭირდებათ რამდენიმე Arduino და ბიუჯეტი შეზღუდული გაქვთ.**
- **მუშაობთ სამაკეტო დაფაზე** და გინდათ, რომ მთელი სისტემა მოათავსოთ მასზე.
- **გჭირდებათ 8 ანალოგური პინი** (Uno-ს 6-ის ნაცვლად).

საბოლოოდ, ფუნქციონალით, ისინი თითქმის **იდენტურნი არიან, რადგან ერთი და იგივე "ტვინი" (ATmega328P) აქვთ. არჩევანი სრულიად დამოკიდებულია თქვენი **პროექტის ფორმაზე, ზომასა და ბიუჯეტზე.**

Bluetooth HC-05 მოდული

რაც შეეხება Bluetooth HC-05 მოდულს. **HC-05** არის კლასიკური, ორმხრივი (Full-Duplex) Bluetooth მოდული, რომელიც მუშაობს **SPP (Serial Port Profile)** პროტოკოლზე. ეს ნიშნავს, რომ ის იქცევა როგორც უსადენო სერიული პორტი (Serial Port). Arduino-ს პროგრამასთან კომუნიკაცია ხდება ზუსტად ისე, როგორც სერიული მონიტორინგისას, მხოლოდ კაბელის ნაცვლად Bluetooth-ით (იხ. სურათი 2).



სურათი 2. Bluetooth HC-05 მოდული

ძირითადი მახასიათებლები და ტექნიკური პარამეტრები:

- **ბლუთუზის ვერსია:** 2.0 + EDR (Enhanced Data Rate)
- **სამუშაო მოდულები:** ორი ძირითადი რეჟიმი
    1. **AT ბრძანებების რეჟიმი:** კონფიგურაციისთვის (სახელის, პაროლის, სიჩქარის შეცვლა).
    2. **მონაცემთა გადაცემის რეჟიმი:** ნორმალური მუშაობა.
- **სიჩქარე (Baud Rate):** ჩვეულებრივ არის 9600, მაგრამ შეიძლება შეიცვალოს (1200, 2400, ..., 115200 და ა.შ.).
- **სამუშაო ძაბვა:** 3.3V - 5V (მნიშვნელოვანია: ზოგიერთ მოდელს აქვს ლოგიკური დონე 3.3V, ამიტომ Arduino-ს 5V პინთან დაკავშირებისას საჭიროა ძაბვის დივაიდერი).
- **დიაპაზონი:** დაახლოებით 10 მეტრი ღია სივრცეში.
- **პაროლი:** ჩვეულებრივ არის `1234` (შეგიძლიათ შეცვლა).
- **სახელი:** `HC-05` (შეგიძლიათ შეცვლა).

როგორ მუშაობს?

ძირითადი იდეა ძალიან მარტივია:

1. თქვენ უკავშირებთ HC-05-ს თქვენს Arduino-ს (RX, TX, VCC, GND პინებით).
2. Arduino-ში ატვირთავთ პროგრამას, რომელიც იყენებს სერიულ კომუნიკაციას.
3. HC-05-ს აწყვილებთ თქვენს ტელეფონთან ან კომპიუტერთან.
4. აპლიკაციიდან ან პროგრამიდან (რომელიც, ასევე, მუშაობს როგორც სერიული პორტი) თქვენ აგზავნით მონაცემებს HC-05-ზე, ის გადასცემს მას Arduino-ს და პირიქით.

მაგალითი: შეგიძლიათ შექმნათ აპლიკაცია ტელეფონზე, სადაც ღილაკის დაჭერით ანიჭებთ ბრძანებას Arduino-ს, რომ ჩართოს/გამორთოს LED, ან მიიღოთ მონაცემები ტემპერატურის სენსორიდან...

გამოყენების სფეროები (რისთვის არის შესაფერისი?)

- **რობოტების მართვა:** ტელეფონიდან რობოტის მართვა.
- **სენსორების მონაცემების გადაცემა:** ტემპერატურა, ტენიანობა, GPS კოორდინატების გაგზავნა ტელეფონზე.
- **სმარტფონთან ინტერფეისი:** Arduino-ს პროექტების დაკავშირება Android ან iOS აპებთან.

- **უსადენო კონტროლი:** სახლის ავტომატიზაციის პროექტები.

HC-05 თუ HC-06

ხშირად ჩნდება კითხვა: რა განსხვავებაა HC-05-სა და HC-06-ს შორის?

- **HC-05:** შეიძლება იყოს როგორც **Master** (იწყებს კავშირს), ისე **Slave** (ელოდება კავშირს). უფრო მრავალფუნქციურია.
- **HC-06:** მხოლოდ **Slave** რეჟიმში მუშაობს (მხოლოდ ელოდება, რომ მას დაუკავშირდნენ). უფრო იაფი და მარტივია.

დასკვნა: HC-05 არის შესანიშნავი, მრავალმხრივი და იაფი ბლუთუზ მოდული, რომელიც იდეალურია Arduino-ს პროექტებისთვის უსადენო კომუნიკაციისთვის. მისი სიმარტივე და SPP პროტოკოლი მას პოპულარულს ხდის დამწყებთა და პროფესიონალთა შორის.

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino Nano
ENA	D5
IN1	D7
IN2	D8
IN3	D9
IN4	D10
ENB	D6

2. დააკავშირეთ ერთმანეთთან არდუინო და Bluetooth HC-05 მოდული:

Bluetooth HC-05	Arduino Nano
TX	RX (Pin 0)
RX	TX (Pin 1)
VCC	+5V
GND	GND

3. დააკავშირეთ ერთმანეთთან არდუინო და LED ნათურები:

2 LED ნათურა	Arduino Nano
დადებითი ფეხი	D11
უარყოფითი ფეხი	GND
დადებითი ფეხი	D12
უარყოფითი ფეხი	GND

4. დააკავშირეთ ერთმანეთთან არდუინო და პიეზო დინამიკი:

პიეზო დინამიკი	Arduino Nano
დადებითი ფეხი	D13
უარყოფითი ფეხი	GND

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

5. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი, ბლუთუზი, LED ნათურები და პიეზო დინამიკი (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino Nano	სამაკეტო დაფა	ძრავას დრაივერი L298N	სამაკეტო დაფა	Bluetooth HC-05	სამაკეტო დაფა	LED	სამაკეტო დაფა	პიეზო	სამაკეტო დაფა
5V	+	+5V	+	VCC	+	+	+	+	+
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND

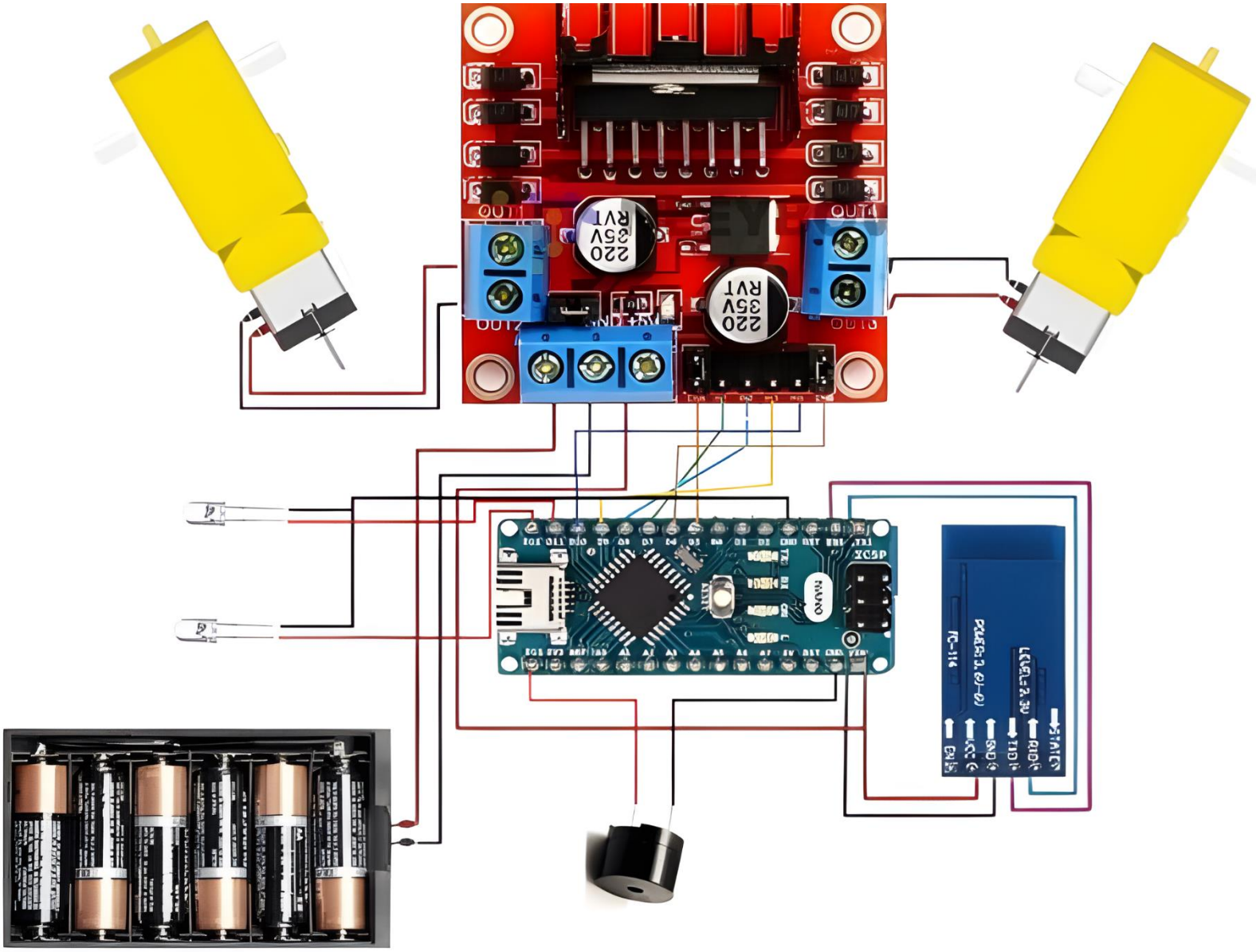
6. დააკავშირეთ არდუინო კომპიუტერთან. კოდის ჩატვირთვამდე Bluetooth მოდულის RX და TX საკონტაქტოები გათიშეთ (ამოიღეთ გამტარები დაფის საკონტაქტო ხვრელებიდან).

7. დააკავშირეთ არდუინო კომპიუტერთან.

> **შეხსენება!**
>
> კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ Arduino NANO და პორტი.

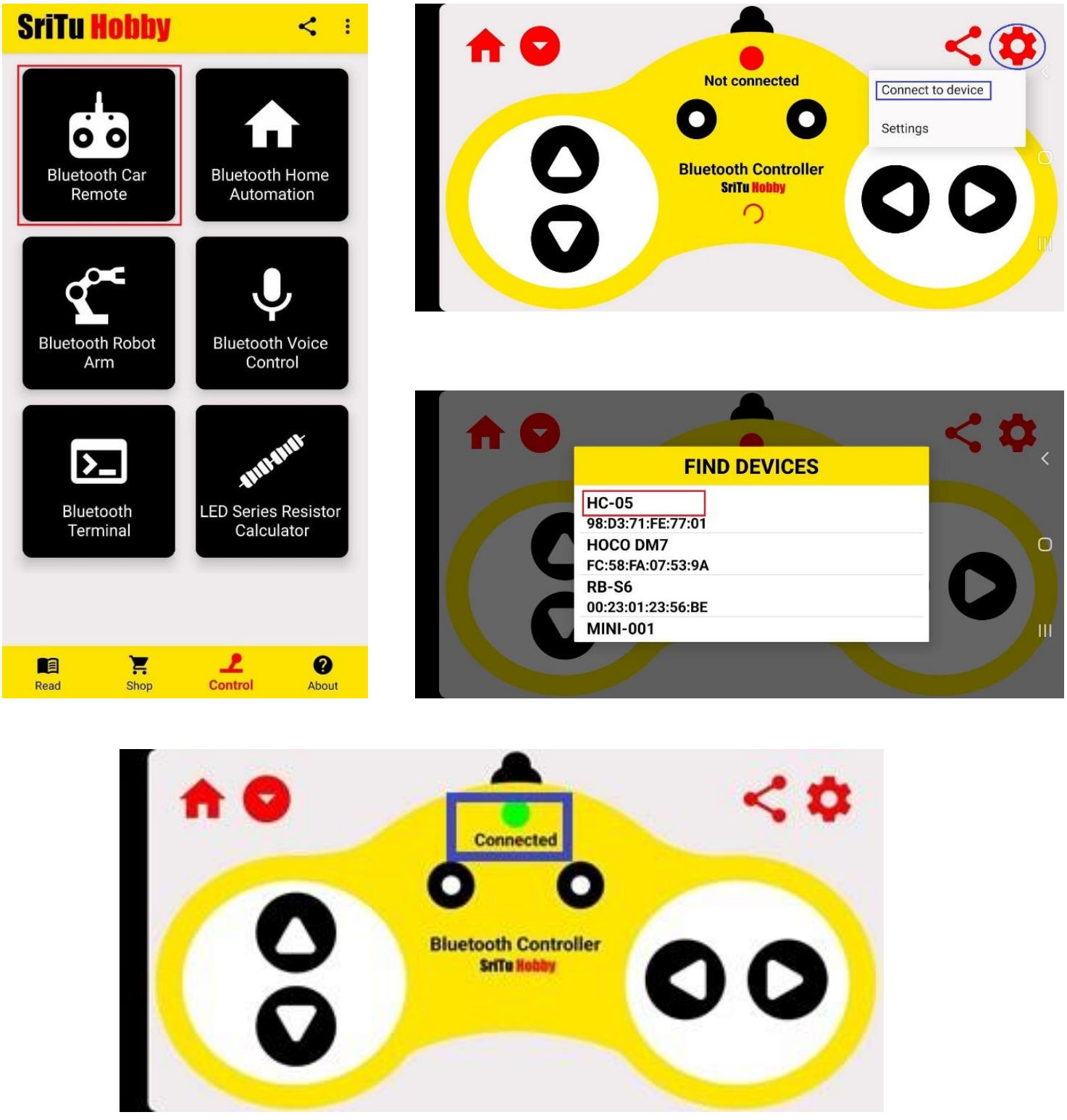
8. ჩატვირთეთ კოდი და კვლავ დააბრუნეთ გამტარები RX და TX კონტაქტებთან.

9. გამორთეთ არდუინო კომპიუტერიდან. მოათავსეთ ელემენტები ბუდეში.



სურათი 3. სქემა

10. Play Store-დან ჩამოტვირთეთ ტელეფონში აპლიკაცია SriTu Hobby. გახსენით აპლიკაცია, გადადით კონტროლერზე და აირჩიეთ ბლუთუზის დისტანციური მართვის პულტი. დააჭირეთ ღილაკს Connect to device და აირჩიეთ თქვენი ბლუთუზის სახელი HC-05. დისტანციური მართვის პულტზე დაინახავთ მწვანე ინდიკატორს (იხ. სურათი 4).



სურათი 4. SriTu Hobby აპლიკაცია და მისი დისტანციური მართვის პულტი

ფარებს (LED ნათურებს) და სიგნალს (პიეზო) მართავს ორი მრგვალი ღილაკი: მარცხენა - ფარებს, მარჯვენა - სიგნალს.

ახლა კი შეგიძლიათ მართოთ თქვენი ავტომობილი.

!!! მომავალში ჩვენ გასწავლით საკუთარი აპლიკაციების შექმნას.

კოდის მუშაობის პრინციპი

1. **Servo Motor**: გამოიყენება ულტრაბგერითი სენსორის მიმართულების შესაცვლელად.
2. **Ultrasonic Sensor (HC-SR04)**: ზომავს მანძილს დაბრკოლებამდე.
3. **L298N Motor Driver**: მართავს ძრავებს.
4. **LEDs და Buzzer**: გამოიყენება განათებისთვის და სიგნალისთვის.
5. **Bluetooth Module**: საშუალებას აძლევს მომხმარებელს, მართოს მანქანა ბლუთუზის მეშვეობით.

ძირითადი ფუნქციები:

- **setup()**: ასრულებს საწყის კონფიგურაციას (pin modes, servo attachment, და სხვა).
- **loop()**: მთავარი ციკლი, რომელიც ამუშავებს bluetoothControl() ფუნქციას.
- **distance()**: ზომავს მანძილს ულტრაბგერითი სენსორის გამოყენებით.
- **obstacle()**: აღიქვამს დაბრკოლებებს და რეაგირებს მათზე (თუ დაბრკოლება არის 9 სმ-ზე ნაკლებ მანძილზე, მანქანა ჯერ უკან დაიხევს და შემდეგ გადაწყვეტს, მარცხნივ უნდა შემობრუნდეს თუ მარჯვნივ).
- **bluetoothControl()**: ამუშავებს ბლუთუზის მეშვეობით მიღებულ ბრძანებებს (მაგ., 'U' - წინ, 'D' - უკან, 'L' - მარცხნივ, 'R' - მარჯვნივ, 'S' - stop, '1'/'2' - LED-ების ჩართვა/გამორთვა, '3'/'4' - ზუმერის ჩართვა/გამორთვა).
- **მოძრაობის ფუნქციები**: Forward(), Backward(), Stop(), Left(), Right().
- **start()**: Servo ძრავას კალიბრაციის ფუნქცია.
- **leftsee() და rightsee()**: ზომავს მანძილს მარცხნივ და მარჯვნივ Servo ძრავას მობრუნებით.

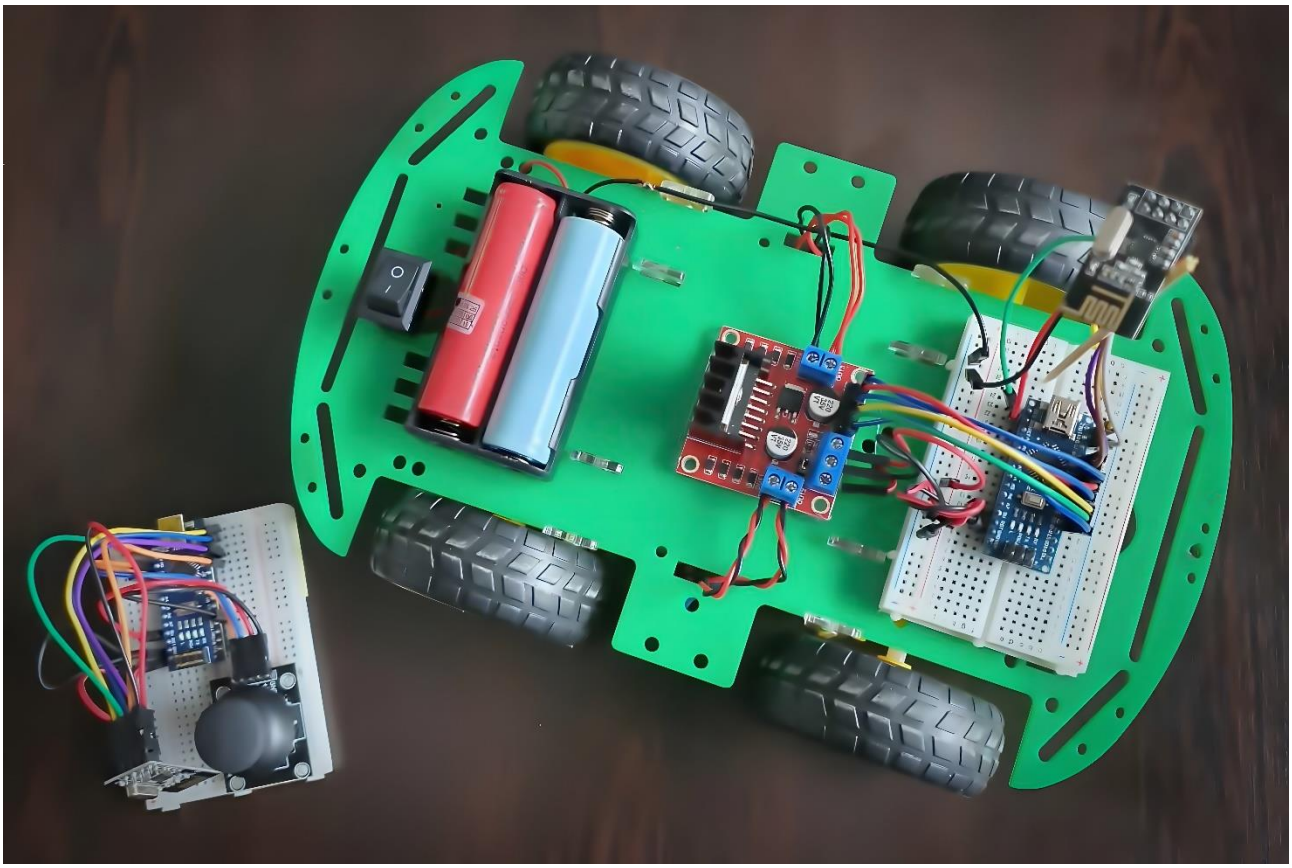
როგორ მუშაობს კოდი:

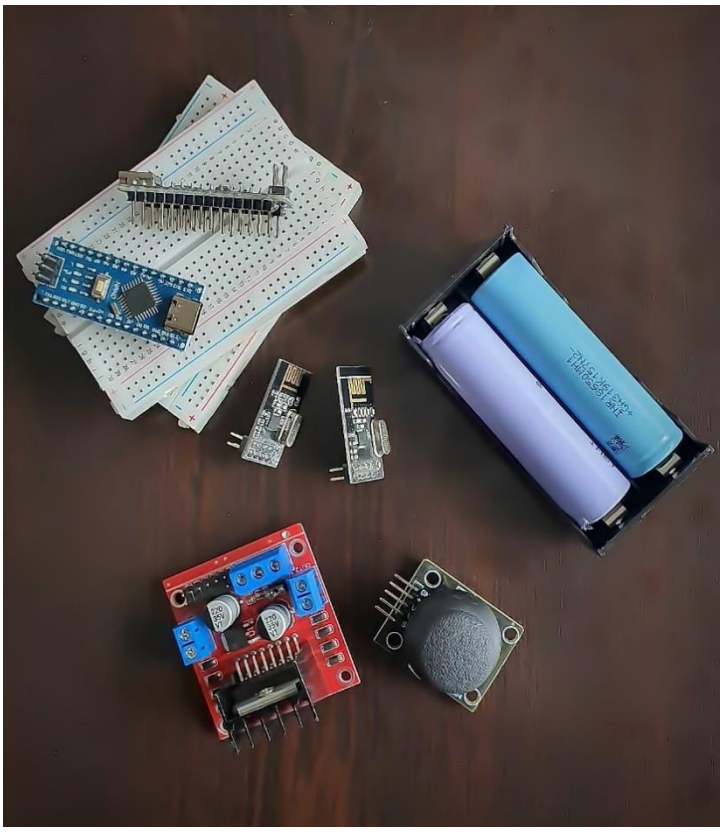
საწყისი კონფიგურაცია: setup()-ში იწყება სერიული კომუნიკაცია, კონფიგურირდება pin-ები და Servo ძრავა, შემდეგ გამოიძახება start() ფუნქცია, რომელიც აკალიბრებს Servo-ს.

მთავარი ციკლი: loop()-ში გაშვებულია bluetoothControl(), რომელიც ამოწმებს სერიულ პორტზე მოსულ მონაცემებს და რეაგირებს შესაბამისად (მაგ., თუ მივიღებთ 'U', მანქანა წინ წავა).

დაბრკოლების აღმოჩენა: obstacle() ფუნქცია ამოწმებს მანძილს და თუ დაბრკოლება აღმოჩნდება, მანქანა უკან დაიხევს და გადაწყვეტს, რომელი მიმართულებით შემობრუნდეს.

რობოტი მანქანის მართვა ჯოისტიკით





საჭირო რესურსი

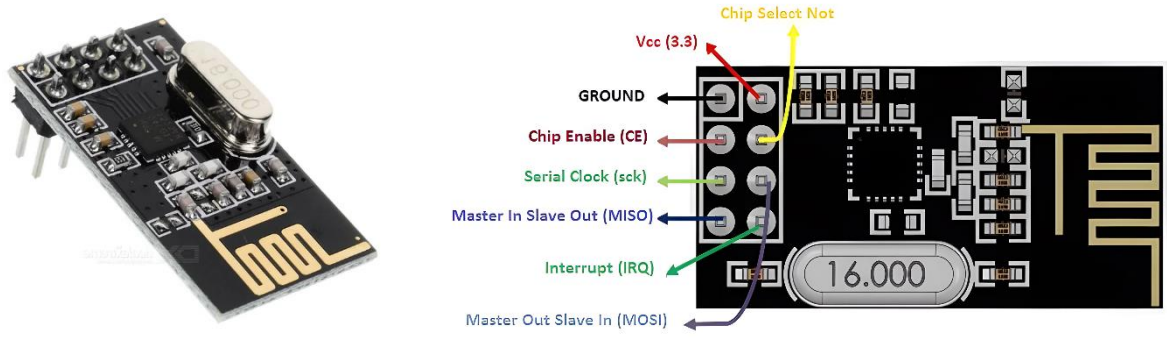
- 4 WD მანქანის კომპლექტი
- 2 ცალი Arduino NANO
- ძრავას დრაივერი L298N
- 2 ცალი NRF24L01 უსადენო რადიოგადამცემი
- 2 ცალი სამაკეტო დაფა
- 4 ცალი ლითიუმის ელემენტი 3.7V და ბუდე
- ორღერძიანი ჯოისტიკის 5-კონტაქტიანი ანალოგური მოდული
- გამტარები: მამალ-მამალი და დედალ-მამალი

საჭირო ბიბლიოთეკა

- RF24

როგორ მუშაობს NRF24L01 უსადენო რადიოგადამცემი

ჩვენ ამ პროექტში გამოვიყენებთ NRF24L01 უსადენო რადიოგადამცემს რობოტის ჯოისტიკით მართვისთვის. ეს გულისხმობს, რომ ჯოისტიკიდან მიღებული სიგნალები NRF24L01-ის საშუალებით (იხ. სურათი 1) გადაეცემა რობოტის მიკროკონტროლერს, რომელიც აკონტროლებს ძრავებს და სხვა ფუნქციებს.



სურათი 1. NRF24L01 უსადენო რადიოგადამცემი

NRF24L01 მოდულს აქვს 8 პინი, რომლებიც გამოიყენება არდუინოსთან კავშირისთვის. ქვემოთ მოცემულია NRF24L01 მოდულის პინების აღწერა:

NRF24L01 პინები:

1. **GND** (Ground):
    - ეს არის "მიწის" პინი, რომელიც უკავშირდება მიკროკონტროლერის GND-ს. ის უზრუნველყოფს საერთო ენერგიის მიწოდებას.

2. **VCC** (Power Supply):
    - ეს პინი უზრუნველყოფს ენერგიის მიწოდებას მოდულისთვის. NRF24L01 მუშაობს 3.3V ძაბვაზე, ამიტომ ფრთხილად უნდა იყოთ, რომ არ მიაწოდოთ 5V, რადგან ამან შეიძლება დააზიანოს მოდული.

3. **CE** (Chip Enable):
    - ეს პინი აკონტროლებს მოდულის აქტიურ/პასიურ რეჟიმს. ის გამოიყენება მოდულის გააქტიურების ან გამორთვისთვის. ჩვეულებრივ, ის უკავშირდება მიკროკონტროლერის დიგიტალურ პინს.

4. **CSN** (Chip Select Not):
    - ეს პინი გამოიყენება SPI ინტერფეისის არჩევისთვის. როდესაც CSN დაბალია (LOW), მოდული აქტიურია SPI კომუნიკაციისთვის. ის ასევე უკავშირდება მიკროკონტროლერის დიგიტალურ პინს.
5. **SCK** (Serial Clock):
    - ეს არის SPI ინტერფეისის საათის პინი (Clock). ის უზრუნველყოფს სინქრონიზაციას მონაცემთა გადაცემის დროს. ის უკავშირდება მიკროკონტროლერის SCK პინს.

6. **MOSI** (Master Out Slave In):
    - ეს პინი გამოიყენება მონაცემთა გადაცემისთვის მიკროკონტროლერიდან NRF24L01 მოდულში. ის უკავშირდება მიკროკონტროლერის MOSI პინს.

7. **MISO** (Master In Slave Out):
    - ეს პინი გამოიყენება მონაცემთა მიღებისთვის NRF24L01 მოდულიდან მიკროკონტროლერისთვის. ის უკავშირდება მიკროკონტროლერის MISO პინს.

8. **IRQ** (Interrupt Request):
    - ეს პინი არის შეწყვეტის (Interrupt) პინი, რომელიც აცნობებს მიკროკონტროლერს მოდულის მდგომარეობის შეცვლის შესახებ (მაგ., მონაცემების მიღების ან გადაცემის დასრულების შესახებ). ის არ არის სავალდებულო გამოსაყენებლად, მაგრამ სასარგებლოა ეფექტური კომუნიკაციისთვის.

NRF24L01 მოდულის ARDUINO-სთან კავშირი:

NRF24L01 მოდული ჩვეულებრივ უკავშირდება Arduino-ს შემდეგი წესით:

- **GND** -> Arduino GND
- **VCC** -> Arduino 3.3V (არასოდეს 5V!)
- **CE** -> Arduino დიგიტალური პინი (მაგ., D9)
- **CSN** -> Arduino დიგიტალური პინი (მაგ., D10)
- **SCK** -> Arduino SCK (D13)
- **MOSI** -> Arduino MOSI (D11)
- **MISO** -> Arduino MISO (D12)
- **IRQ** -> Arduino დიგიტალური პინი (არასავალდებულო, მაგ., D8)

მნიშვნელოვანი შენიშვნები:

- **3.3V ძაბვა:** NRF24L01 მოდული მგრძნობიარეა ძაბვის მიმართ. დარწმუნდით, რომ მხოლოდ 3.3V-ს მიაწოდებთ VCC პინს. 5V-ის მიწოდება დააზიანებს მოდულს.
- **ენერგიის სტაბილურობა:** რეკომენდებულია 10-100 µF კონდენსატორის გამოყენება VCC და GND პინებს შორის, რათა თავიდან აიცილოთ ძაბვის ვარდნები.
- **ანტენა** (იმ შემთხვევაში, თუ გამოიყენებთ): უზრუნველყავით, რომ ანტენა სწორად იყოს დაკავშირებული და არ იყოს დაზიანებული.

როგორ მუშაობს პროექტი

1. **გადამცემი (Transmitter)**: ჯოისტიკი უკავშირდება Arduino-ს, რომელიც ამუშავებს ჯოისტიკის სიგნალებს და აგზავნის მათ NRF24L01 მოდულის საშუალებით.
2. **მიმღები (Receiver):** რობოტზე დამონტაჟებული NRF24L01 მოდული იღებს სიგნალებს და გადასცემს მათ რობოტის მიკროკონტროლერს, რომელიც აკონტროლებს ძრავებს და სხვა კომპონენტებს.

კავშირის დიაგრამა:

- გადამცემი მხარე (ჯოისტიკი):
    - ჯოისტიკი -> Arduino -> NRF24L01
- მიმღები მხარე (რობოტი):
    - NRF24L01 -> Arduino -> ძრავების დრაივერი -> ძრავები

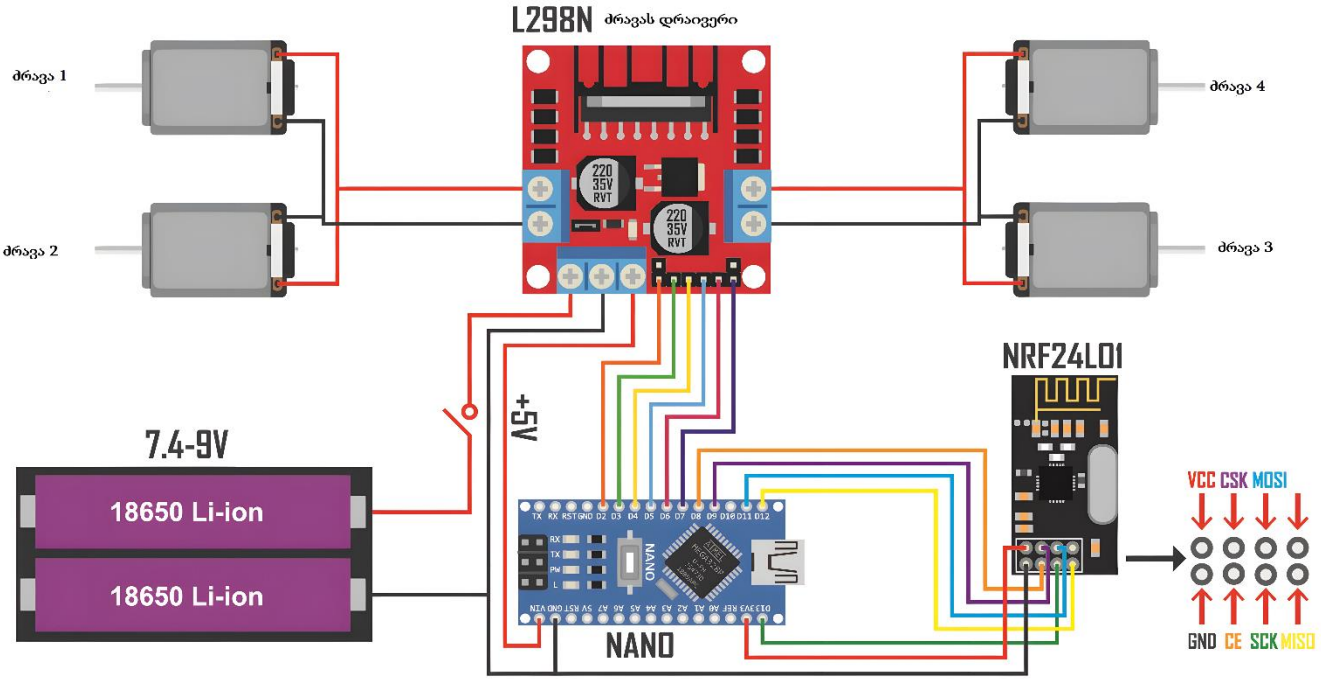
გაითვალისწინეთ!

ამ პროექტისთვის ჩვენ დაგვჭირდება ორი სქემა და შესაბამისად, ორი კოდი:

1. მიმღების სქემა;
2. გადამცემის სქემა.

ავაწყოთ I სქემა

1. ჯოისტიკის მიმღების სქემა (ARDUINO JOYSTICK RECEIVER SCHEMATIC - ამ დასახელებით იქნება კოდი)



სურათი 2. მიმღების სქემა (4WD რობოტი)

1. დააკავშირეთ ერთმანეთთან არდუინო და L298N ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino NANO
ENA	Pin 2
IN1	Pin 3
IN2	Pin 4
IN3	Pin 5
IN4	Pin 6
ENB	Pin 7

2. დააკავშირეთ ერთმანეთთან არდუინო და NRF24L01 უსადენო რადიოგადამცემის მოდული:

NRF24L01 მოდული	Arduino NANO
GND	GND
CE	Pin 8
SCK	Pin 13
MISO	Pin 12
VCC	3V
CSK	Pin 9
MOSI	Pin 11

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

3. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი L298N და NRF24L01 მოდული (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino NANO	სამაკეტო დაფა
5V	+
GND	GND

ძრავას დრაივერი	სამაკეტო დაფა
5V	+
GND	GND

NRF24L01 მოდული	სამაკეტო დაფა
VCC	+
GND	GND

4. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ Arduino NANO და პორტი.

5. ჩატვირთეთ კოდი.

ვიდრე ჯოისტიკის გადამცემის სქემის აწყობას დაიწყებთ, ცოტა რამ უნდა იცოდეთ ჯოისტიკის შესახებ (სურათი 3). ჯოისტიკი, ძირითადად, შედგება ორი პოტენციომეტრისგან და ღილაკისგან, რაც საშუალებას გვაძლევს გავზომოთ მოძრაობა ორ განზომილებაში.

სურათი 3. ორღერძიანი ჯოისტიკის 5-კონტაქტიანი ანალოგური მოდული

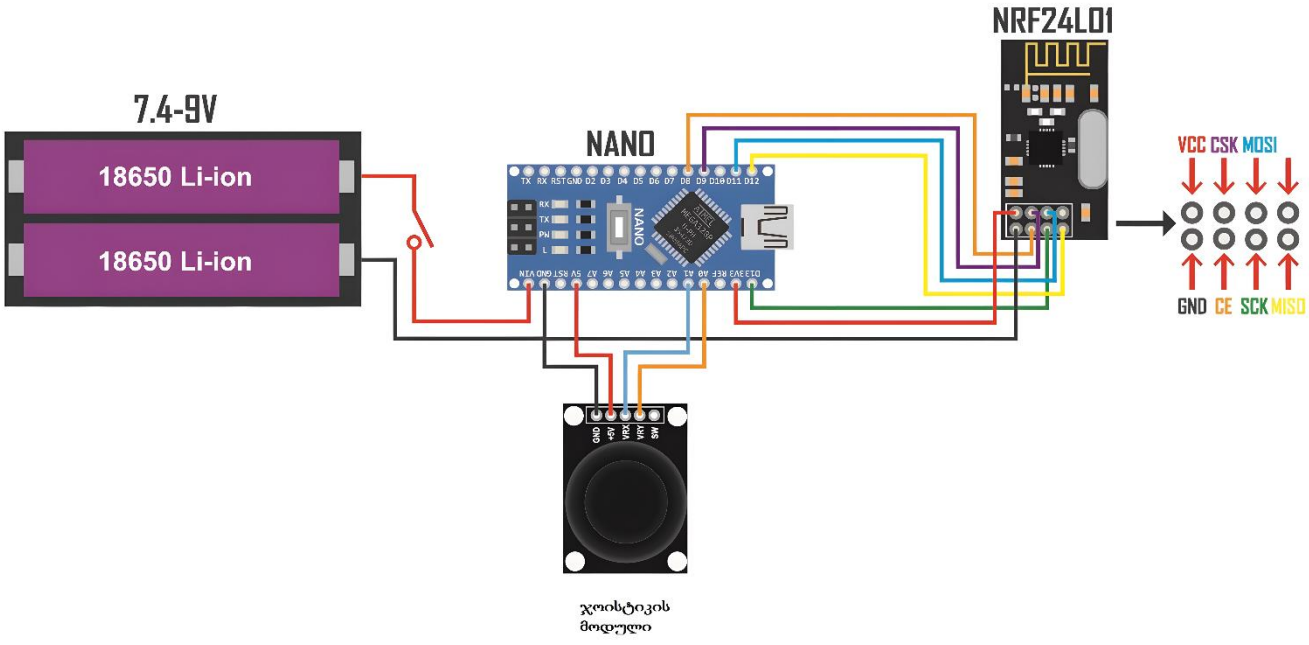
ორღერძიანი ჯოისტიკის მოდულის პინები:

1. **GND** (Ground):
   - ეს არის "მიწის" პინი, რომელიც უკავშირდება მიკროკონტროლერის (მაგ., Arduino) GND-ს. ის უზრუნველყოფს საერთო ენერგიის მიწოდებას.
2. **+5V** (VCC):
   - ეს პინი უზრუნველყოფს ენერგიის მიწოდებას ჯოისტიკისთვის. ის ჩვეულებრივ უკავშირდება Arduino-ს 5V პინს.

3. **VRx** (X-Axis):
    - ეს პინი არის X-ღერძის (ჰორიზონტალური მოძრაობის) ანალოგური სიგნალის გამომავალი. ის უკავშირდება Arduino-ს ანალოგურ პინს (მაგ., A0) და ცვლის ძაბვის მნიშვნელობას, რომელიც იცვლება ჯოისტიკის X-ღერძის მიმართულებიდან გამომდინარე.
4. **VRy** (Y-Axis):
    - ეს პინი არის Y-ღერძის (ვერტიკალური მოძრაობის) ანალოგური სიგნალის გამომავალი. ის ასევე უკავშირდება Arduino-ს ანალოგურ პინს (მაგ., A1) და ცვლის ძაბვის მნიშვნელობას, რომელიც იცვლება ჯოისტიკის Y-ღერძის მიმართულებიდან გამომდინარე.
5. **SW** (Switch) - ამ პინს არ ვიყენებთ:
    - ეს პინი არის ჯოისტიკის ღილაკის (Button) სიგნალის გამომავალი. როდესაც ჯოისტიკის ღილაკი დაჭერილია, ეს პინი დაბალ დონეზე (LOW) გადადის. ის ჩვეულებრივ უკავშირდება Arduino-ს დიგიტალურ პინს (მაგ., D2).

ავაწყოთ II სქემა

ჯოისტიკის გადამცემის სქემა (ARDUINO JOYSTICK TRANSMITTER SCHEMATIC - ამ დასახელებით იქნება კოდი)



სურათი 4. გადამცემის სქემა

1. დააკავშირეთ არდუინო და NRF24L01 უსადენო რადიოგადამცემის მოდული:

NRF24L01 მოდული	Arduino NANO
GND	GND
CE	Pin D8
SCK	Pin D13
MISO	Pin D12
VCC	3V
CSK	Pin D9
MOSI	Pin D11

2. დააკავშირეთ ერთმანეთთან არდუინო და ჯოისტიკის მოდული:

ჯოისტიკის მოდული	Arduino NANO
GND	GND
5V	5V
VRX	Pin A1
VRY	Pin A0

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

3. დააკავშირეთ ერთმანეთთან არდუინო, NRF24L01 და ჯოისტიკის მოდულები სამაკეტო დაფაზე (ვინაიდან ამ პროექტის კავშირებისთვის არდუინოს 5V-სა და GND-ს პინები არ არის საკმარისი, ამიტომ ვიყენებთ სამაკეტო დაფას).

Arduino NANO	სამაკეტო დაფა
5V	+
GND	GND

NRF24L01 მოდული	სამაკეტო დაფა
VCC	+
GND	GND

ჯოისტიკის მოდული	სამაკეტო დაფა
5V	+
GND	GND

4. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools, მონიშნეთ Arduino NANO და პორტი.

5. ჩატვირთეთ კოდი.

კოდის მუშაობის პრინციპი

ეს კოდი აკონტროლებს მანქანას, რომელიც მართავს ჯოისტიკით უსადენოდ (nRF24L01 მოდულის გამოყენებით). კოდი იყენებს L298N ძრავის დრაივერს. ჯოისტიკის X და Y ღერძები განსაზღვრავენ მანქანის მიმართულებას და სიჩქარეს.

1. კოდის ძირითადი კომპონენტები

- **nRF24L01 მოდული:** გამოიყენება უსადენო კომუნიკაციისთვის. ის იღებს მონაცემებს ჯოისტიკიდან.
- **L298N დრაივერი:** აკონტროლებს ძრავების მიმართულებას და სიჩქარეს.
- **ჯოისტიკი:** აგზავნის X და Y ღერძების მონაცემებს, რომლებიც განსაზღვრავენ მანქანის მოძრაობას.

2. კოდის მუშაობის ეტაპები

❖ **ინიციალიზაცია (setup()):**

- პინები (enA, in1, in2, enB, in3, in4) კონფიგურირებულია, როგორც გამომავალი (OUTPUT), რათა მართონ ძრავები.
- nRF24L01 მოდული ინიციალიზირებულია და კონფიგურირებულია მიმღების რეჟიმში (startListening).

❖ **მონაცემების მიღება (loop()):**

- თუ nRF24L01 მოდულზე არის მიღებული მონაცემები (radio.available()), კოდი კითხულობს მონაცემებს (radio.read()).
- მიღებული მონაცემები ინახება receivedData მასივში.
- X და Y ღერძების მნიშვნელობები გადაიყვანება რიცხვებად (atoi ფუნქციით) და ინახება xAxis და yAxis ცვლადებში.

❖ **ძრავების კონტროლი**

- **Y-ღერძი (წინ და უკან):**
  - თუ yAxis მნიშვნელობა 470-ზე ნაკლებია, მანქანა მოძრაობს **უკან**:
    - in1 და in3 პინები გადადის HIGH მდგომარეობაში, ხოლო in2 და in4 LOW-ში.
    - ძრავების სიჩქარე (motorSpeedA და motorSpeedB) იზრდება map() ფუნქციის გამოყენებით.
  - თუ yAxis მნიშვნელობა 550-ზე მეტია, მანქანა მოძრაობს **წინ**:
    - in2 და in4 პინები გადადის HIGH მდგომარეობაში, ხოლო in1 და in3 LOW-ში.
    - ძრავების სიჩქარე ისევ იზრდება map() ფუნქციით.
  - თუ yAxis მნიშვნელობა 470-დან 550-მდეა, ძრავები **გაჩერებულია**.
- **X-ღერძი (მარცხნივ და მარჯვნივ):**
  - თუ xAxis მნიშვნელობა 470-ზე ნაკლებია, მანქანა მოძრაობს **მარცხნივ**:
    - მარცხენა ძრავას სიჩქარე (motorSpeedA) მცირდება, ხოლო მარჯვენასი (motorSpeedB) - იზრდება.
  - თუ xAxis მნიშვნელობა 550-ზე მეტია, მანქანა მოძრაობს **მარჯვნივ**:
    - მარჯვენა ძრავას სიჩქარე (motorSpeedB) მცირდება, ხოლო მარცხენასი (motorSpeedA) - იზრდება.
  - სიჩქარეები შემოიფარგლება 0-დან 255-მდე, რათა არ მოხდეს გადაჭარბება PWM სიგნალის დიაპაზონის.

❖ **ძრავების გაჩერება დაბალი სიჩქარის დროს**

- თუ ძრავების სიჩქარე 70-ზე ნაკლებია, ისინი გაჩერებულია.

❖ **PWM სიგნალების გამოყენება**

- analogWrite(enA, motorSpeedA); და analogWrite(enB, motorSpeedB); – ეს ფუნქციები აგზავნიან PWM სიგნალებს ძრავებზე, რაც განსაზღვრავს მათ სიჩქარეს.

3. როგორ მუშაობს მანქანა?

- **წინ:** ჯოისტიკი გადაადგილებულია წინ (Y-ღერძი > 550), ძრავები მოძრაობენ წინ.
- **უკან:** ჯოისტიკი გადაადგილებულია უკან (Y-ღერძი < 470), ძრავები მოძრაობენ უკან.
- **მარცხნივ:** ჯოისტიკი გადაადგილებულია მარცხნივ (X-ღერძი < 470), მარცხენა ძრავა ნელა მოძრაობს, ხოლო მარჯვენა სწრაფად.
- **მარჯვნივ:** ჯოისტიკი გადაადგილებულია მარჯვნივ (X-ღერძი > 550), მარჯვენა ძრავა ნელა მოძრაობს, ხოლო მარცხენა სწრაფად.
- **გაჩერება:** ჯოისტიკი ნეიტრალურ მდგომარეობაშია (X და Y ღერძები 470-დან 550-მდე), ძრავები გაჩერებულია.

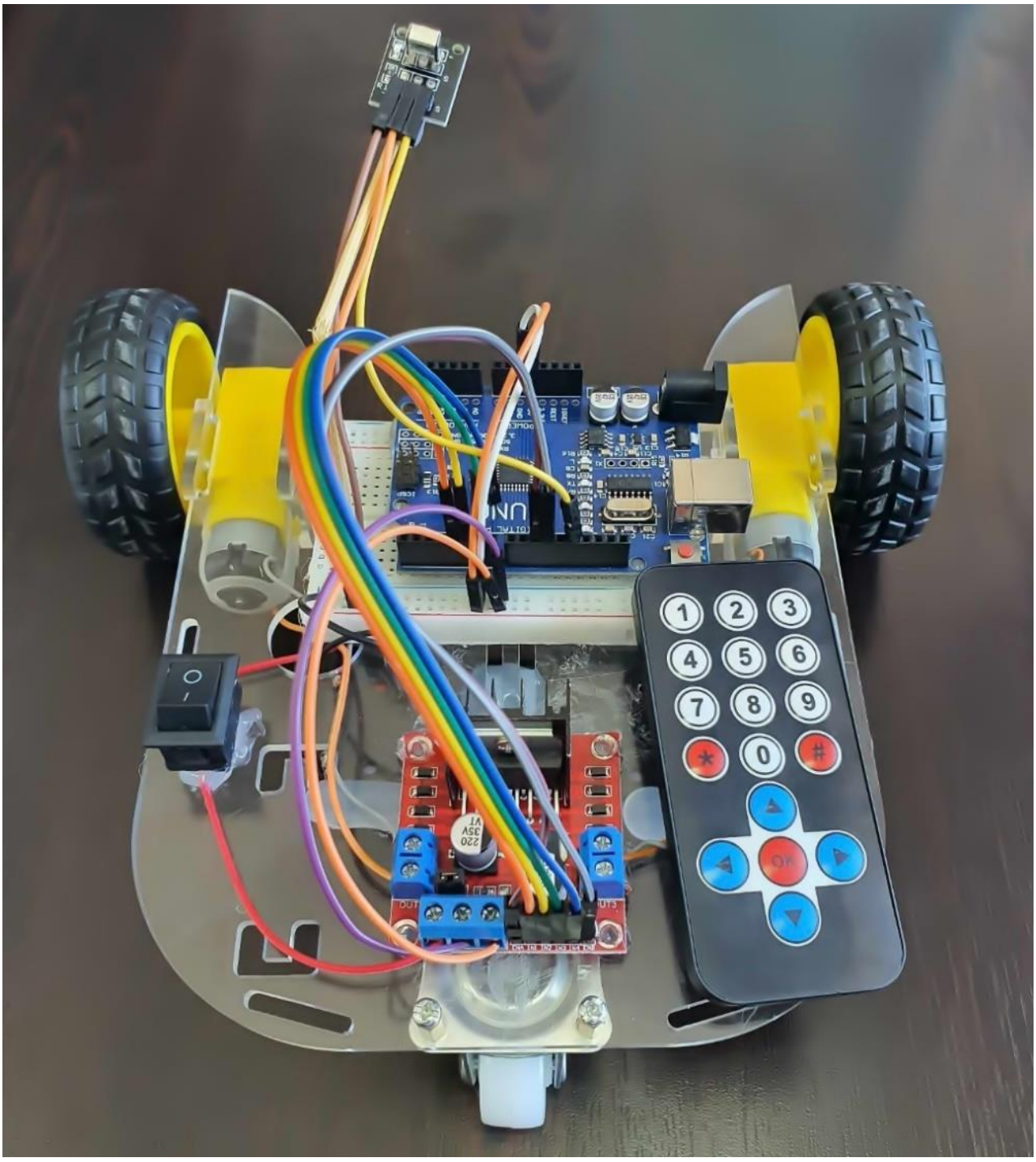
4. რატომ არის ეს კოდი ეფექტური?

- უსადენო კონტროლი: nRF24L01 მოდული უზრუნველყოფს სტაბილურ და სწრაფ კომუნიკაციას.
- PWM სიგნალები: analogWrite() ფუნქცია საშუალებას აძლევს ძრავებს მოძრაობდნენ სხვადასხვა სიჩქარით.
- მარტივი კონფიგურაცია: კოდი ადვილად მოდიფიცირებადია სხვა პროექტებისთვის.

დასკვნა

ეს კოდი არის მარტივი და ეფექტური გადაწყვეტა ჭკვიანი მანქანის უსადენო კონტროლისთვის. ის იყენებს ჯოისტიკის მონაცემებს, რათა განსაზღვროს ძრავების მიმართულება და სიჩქარე, ხოლო nRF24L01 მოდული უზრუნველყოფს კომუნიკაციას მანქანასა და ჯოისტიკს შორის.

რობოტი მანქანა დისტანციური მართვის პულტით



საჭირო ბიბლიოთეკა

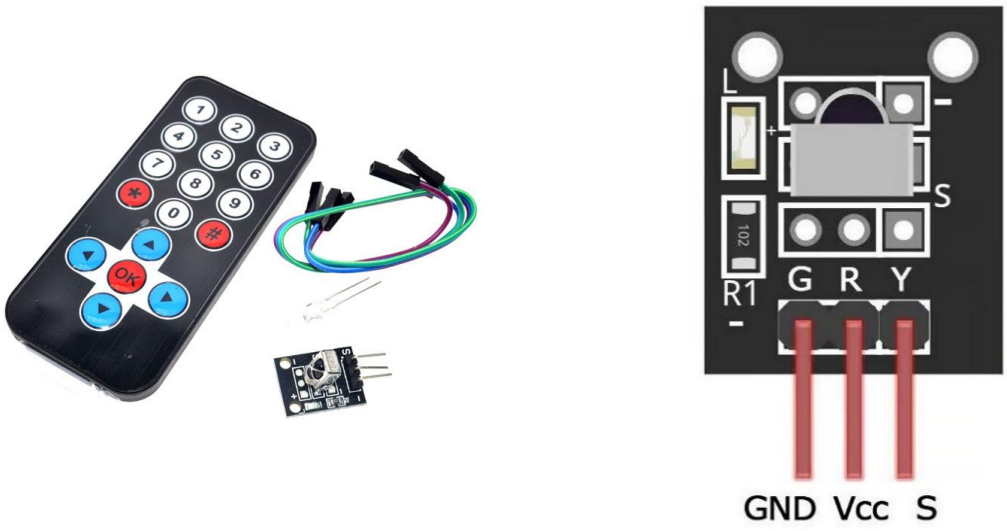
- IRremote

საჭირო რესურსი

- Arduino UNO
- 2WD რობოტი მანქანა (კომპლექტი)
- სამაკეტო დაფა
- გამტარები (მამალ-მამალი და დედალ-მამალი)
- ძრავას დრაივერი L298N
- დისტანციური მართვის პულტი
- ინფრაწითელი (IR) მიმღები
- 2 ცალი ლითიუმის ელემენტი 3.7V და ბუდე

როგორ მუშაობს რობოტი მანქანა

ამ პროექტში ჩვენ შევქმნით 2WD მანქანას, რომელიც იმართება დისტანციური მართვის პულტით + ინფრაწითელი (IR) მიმღები. გაითვალისწინეთ, რომ სქემის აწყობისას თუ ინფრაწითელი მიმღების პინებს სწორად არ დააკავშირებთ, ის მარტივად გადაიწვება (იხ. სურათი 1).



სურათი 1. დისტანციური მართვის პულტი და ინფრაწითელი მიმღების საკონტაქტო პინები.

ინფრაწითელი (IR) მიმღები

ინფრაწითელ (IR) მიმღებს აქვს 3 პინი:

- **VCC (3.3V–5V)**: კვების წყარო.
- **GND**: მიწა.
- **OUT (Signal)**: დემოდულირებული სიგნალის გამოსასვლელი.

როგორ მუშაობს

- **ფოტოდიოდი** იღებს IR სიგნალს და გარდაქმნის მას ელექტრულ იმპულსებად.
- **შიდა დემოდულატორი** ამოწმებს სიგნალს 38 kHz სიხშირეზე.
- **OUT პინი** აგზავნის **დეკოდირებულ** სიგნალს (ჩვეულებრივ TTL დონეზე – LOW/HIGH), რომელიც შეესაბამება პულტის ბრძანებებს.

მიკროკონტროლერთან (არდუინო) მუშაობა:

- OUT სიგნალი უკავშირდება მიკროკონტროლერის (მაგ., Arduino) **დიგიტალურ შესასვლელს**.
- მიკროკონტროლერი იყენებს ბიბლიოთეკებს (მაგ., IRremote.h Arduino-სთვის) სიგნალის დეკოდირებისთვის და ბრძანების გამოსაცნობად.

IR სიგნალის დისტანცია: სტანდარტული ინფრაწითელი (IR) მიმღები ჩვეულებრივ მუშაობს **3-5 მეტრამდე**. თუ მეტი სიშორე გჭირდებათ, გამოიყენეთ გამაძლიერებელი ან **RF (რადიო) მოდული**.

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino UNO
ENA	Pin 11
IN1	Pin 4
IN2	Pin 5
IN3	Pin 6
IN4	Pin 7
ENB	Pin 10

2. დააკავშირეთ ერთმანეთთან არდუინო და ინფრაწითელი (IR) მიმღები:

ინფრაწითელი (IR) მიმღები	Arduino UNO
GND	GND
VCC	+5V
OUT (s)	Pin 13

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

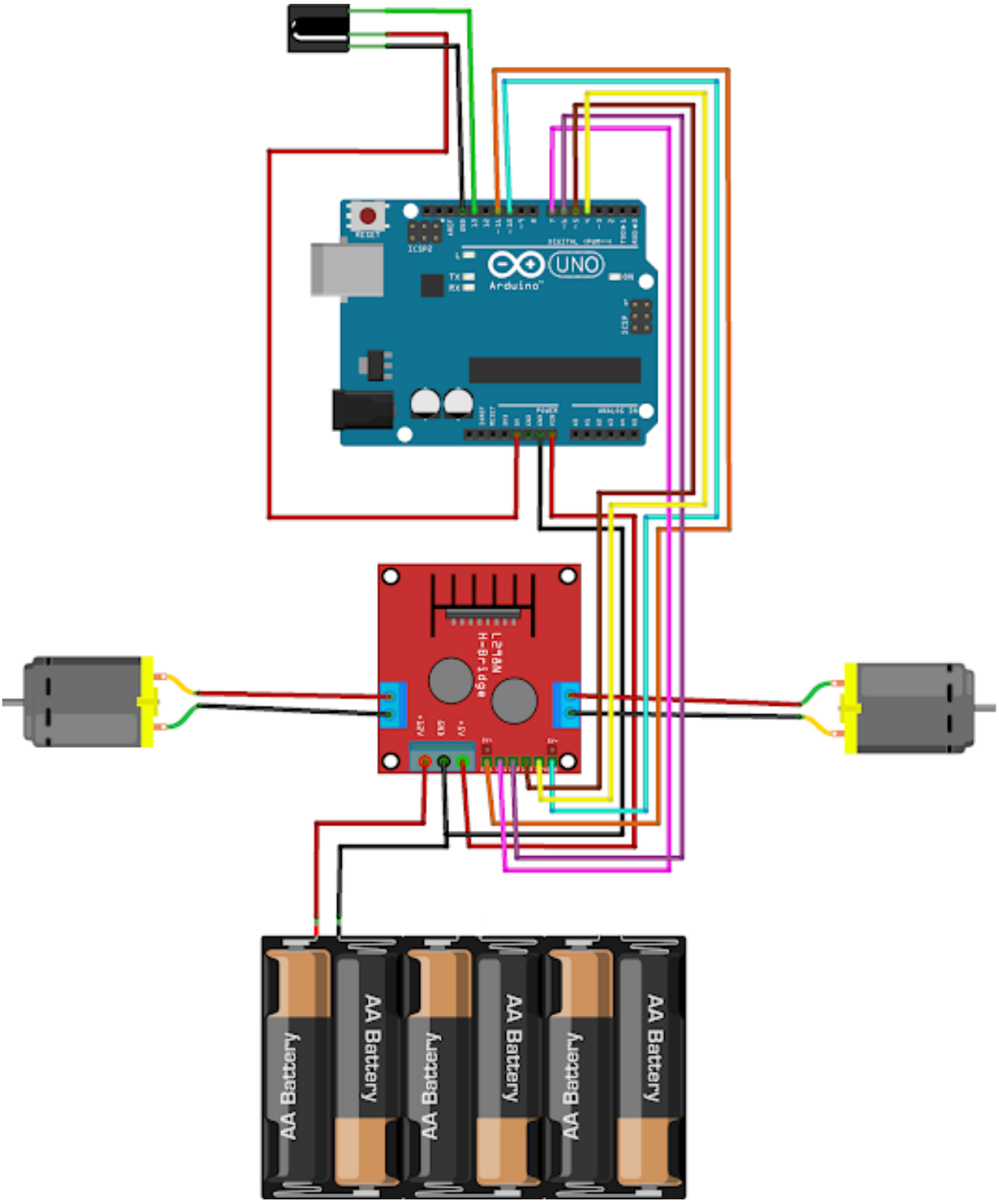
3. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი და ინფრაწითელი (IR) მიმღები (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino UNO	სამაკეტო დაფა
5V	+
GND	GND

ძრავას დრაივერი L298N	სამაკეტო დაფა
+5V	+
GND	GND

ინფრაწითელი (IR) მიმღები	სამაკეტო დაფა
VCC	+
GND	GND

პროექტის სქემა ასე გამოიყურება (იხ. სურათი 2):



სურათი 2. სქემა

დისტანციური მართვის პულტის ღილაკების დაყენება

რობოტი რომ ვმართოთ, ამისათვის საჭიროა დისტანციური მართვის პულტის დეკოდირება ინფრაწითელი მიმღების საშუალებით. როგორც უკვე ვიცით, მიმღებს აქვს სამი გამომყვანი: OUT (s), GND და VCC (იხ. სურათი 1).

დისტანციური მართვის პულტის ღილაკზე დაჭერისას ის აგზავნის ციფრულ მნიშვნელობას, რომელსაც იღებს ინფრაწითელი მიმღები. ეს მნიშვნელობა განსხვავებულია ყველა ღილაკისთვის. ჩვენ გავშიფრავთ თითოეული ღილაკის მნიშვნელობებს Arduino-ს საშუალებით და შემდეგ მივანიჭებთ მათ Arduino-ს პინებს კოდში, რათა გავაკონტროლოთ გამომავალი - ამ შემთხვევაში, ძრავები.

თქვენ მიერ გაშიფრული მნიშვნელობებით, პერსონალიზებული კოდის საშუალებით, შეგიძლიათ დააკავშიროთ გარკვეული ღილაკები გარკვეულ ინსტრუქციებთან და გამოიყენოთ დისტანციური პულტი ძრავების სამართავად.

ჩვენ მივანიჭებთ ღილაკს ძრავების მოძრაობის მიმართულებას. მთლიანობაში, სულ ხუთი ღილაკი გააკონტროლებს ყველა მოძრაობას: წინ და უკან, მარცხნივ და მარჯვნივ და გაჩერება (Stop).

1. ჩამოტვირთეთ IRremote ბიბლიოთეკა და ჩატვირთეთ არდუინოს IDE ინტერფეისში (ბიბლიოთეკების დაყენების ინსტრუქციები იხილეთ: „არდუინოს დამატებითი ბიბლიოთეკების ინსტალაცია“, ვებგვერდი: https://chkhirkedela2019.wixsite.com/mysite/about-5).

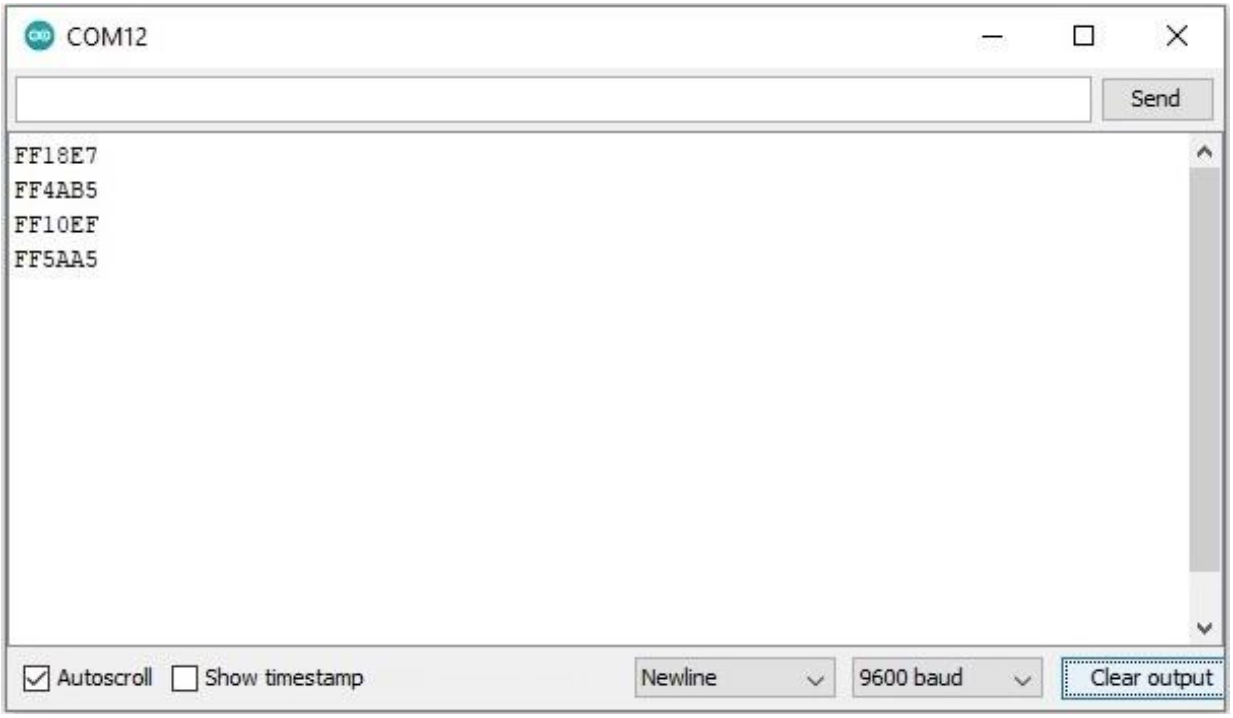
2. ჩატვირთეთ **Project_7a** კოდი (დისტანციური მართვის წამკითხველი).

როგორ მუშაობს დისტანციური მართვის წამკითხველი კოდი

თავდაპირველად კოდი მიმართავს IRremote ბიბლიოთეკას, რომელიც კითხულობს მონაცემებს ინფრაწითელი მიმღებიდან და შესაბამის მონაცემებს აგზავნის არდუინოზე. მიმღებს ვანიჭებთ არდუინოს მე-13 საკონტაქტო პინს და კოდი იწყებს არდუინოს ინტერფეისთან ურთიერთქმედებას, რათა ღილაკზე დაჭერისას შემომავალი მონაცემები რეალურ დროში აისახოს მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor). კოდი მუშაობს ციკლურად და ღილაკზე დაჭერისას გვიჩვენებს მის შესაბამის მნიშვნელობას.

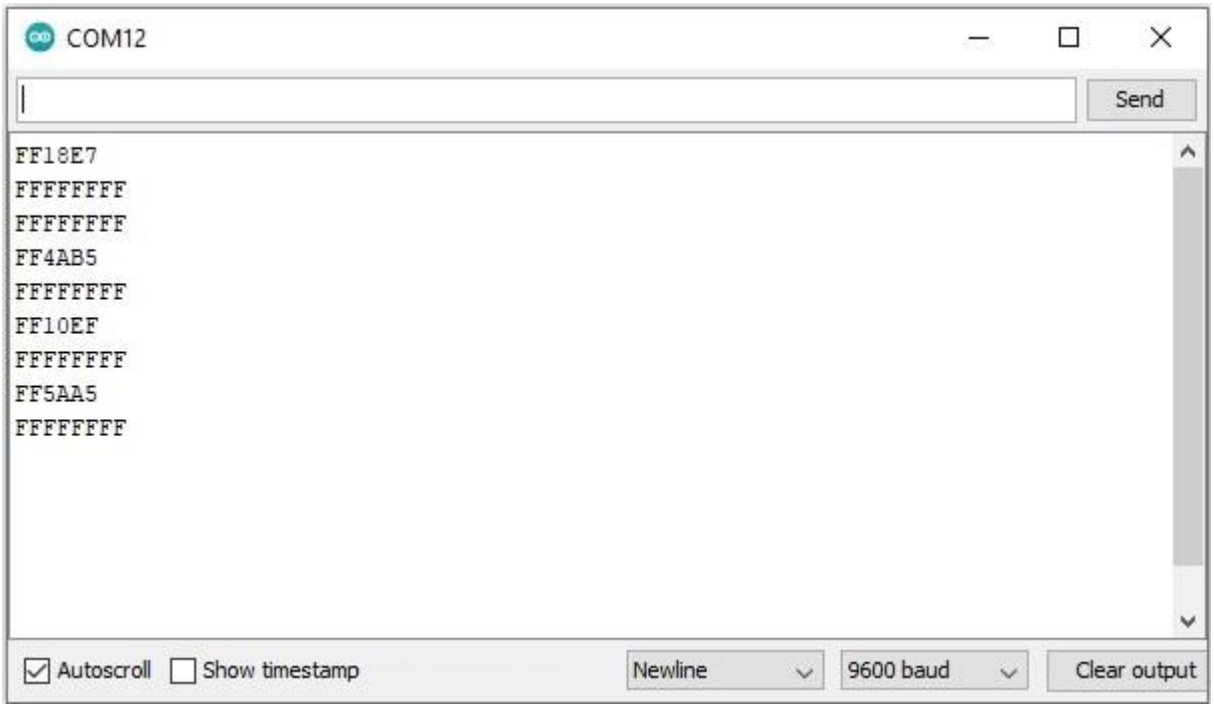
1. გახსენით არდუინოს ინტერფეისში მიმდევრობითი კომუნიკაციის მონიტორი (Serial Monitor).

2. მიმართეთ პულტი მიმღებისკენ და დააჭირეთ სხვადასხვა ღილაკს. მათი დეკოდირებული მნიშვნელობები გამოჩნდება მონიტორზე (იხ. სურათი 3).



სურათი 3. ღილაკების დეკოდირებული მნიშვნელობები

ჩაიწერეთ, რომელ ღილაკს რომელი მნიშვნელობა შეესაბამება. ეს მნიშვნელობები მოგვიანებით დაგჭირდებათ. თუ ღილაკებს ხანგრძლივად დააჭერთ, სურათზე გამოსახული მნიშვნელობების გარდა, სხვა ტექსტსაც დაინახავთ, მაგრამ თქვენ მხოლოდ მესამე სურათზე გამოსახული მსგავსი კონფიგურაციის მნიშვნელობები უნდა ჩაინიშნოთ. ყველა პულტისთვის ეს მნიშვნელობები განსხვავებულია (იხ. სურათი 4).



სურათი 4. როცა ღილაკს ხანგრძლივად ვაჭერთ

ახლა უკვე, როცა გავშიფრეთ დისტანციური მართვის პულტის ღილაკების მნიშვნელობები, ჩვენ შეგვიძლია მათი გამოყენება ძრავების სამართავად.

3. შეამოწმეთ, რომ ზუსტად შეესაბამებოდეს თქვენ მიერ აწყობილი სქემა ჩვენს სქემას და კოდის (**Project_7 b)** ჩატვირთამდე დარწმუნდით, რომ ჩვენი მნიშვნელობები ჩაანაცვლეთ თქვენ მიერ გაშიფრული დისტანციური მართვის პულტის ღილაკების დეკოდირებული მნიშვნელობებით. მნიშვნელობების შეცვლისას ვინარჩუნებთ 0x-ს და მხოლოდ შემდეგ ვამატებთ ჩვენს მონაცემებს. მაგალითად, ჩვენი პირველი ღილაკების დეკოდირებული მნიშვნელობაა FF18E7, კოდში კი არის:

---

if (results.value == 0xFF6897)

---

4. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!
კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ პორტი.

5. ჩატვირთეთ კოდი.

გამორთეთ არდუინო კომპიუტერიდან. ჩაალაგეთ ელემენტები ბუდეში და მართეთ მანქანა.

როგორ მუშაობს კოდი

ეს კოდი აკონტროლებს **რობოტ მანქანას ინფრაწითელი (IR) დისტანციური მართვის გამოყენებით**. ძირითადი ფუნქციები: წინ, უკან, მარცხნივ/მარჯვნივ მობრუნება და გაჩერება.

❖ **ძირითადი კომპონენტები:**

- **IR მიმღები** (RECV_PIN = 13) – იღებს სიგნალებს დისტანციურიდან.
- **DC ძრავების მართვა** (პინები 4, 5, 6, 7, 10, 11) – ძრავას დრაივერის საშუალებით აკონტროლებს ძრავებს.
- **Serial Monitor** – აჩვენებს მოქმედებებს (წინ, უკან, მარჯვნივ, მარცხნივ, stop).

❖ **ფუნქციები:**

- **forward()**
    - ორივე ძრავა მუშაობს **წინ** (მაქსიმალური სიჩქარე: analogWrite(255)).
    - პინები 4 (HIGH) და 5 (LOW) – ერთი ძრავა წინ.
    - პინები 6 (LOW) და 7 (HIGH) – მეორე ძრავა წინ.
- **backwards()**
    - ძრავები მუშაობენ **უკან** (პინების პოლარობა შებრუნებულია).
- **left() და right()**
    - ძრავები მოძრაობენ საპირისპირო მიმართულებით (1 წამი), რაც იწვევს მობრუნებას.
    - **delay(1000)** – განსაზღვრავს მობრუნების დროს.
- **stopp()**
    - ორივე ძრავა **ჩერდება** (ყველა პინი LOW, PWM = 0).

❖ **ინფრაწითელი (IR) დისტანციურ მართვის პულტის დეკოდირება:**

- **loop()** ფუნქცია ამოწმებს დისტანციურიდან მიღებულ სიგნალებს:
    - **0xFF6897** → **Stop** (stopp()).
    - **0xFFA857** → **წინ** (forward() + მცირე დაყოვნება).
    - **0xFFE01F** → **უკან** (backwards() + 2 წამი).
    - **0xFF02FD** → **მარჯვნივ** (right()).
    - **0xFF22DD** → **მარცხნივ** (left()).

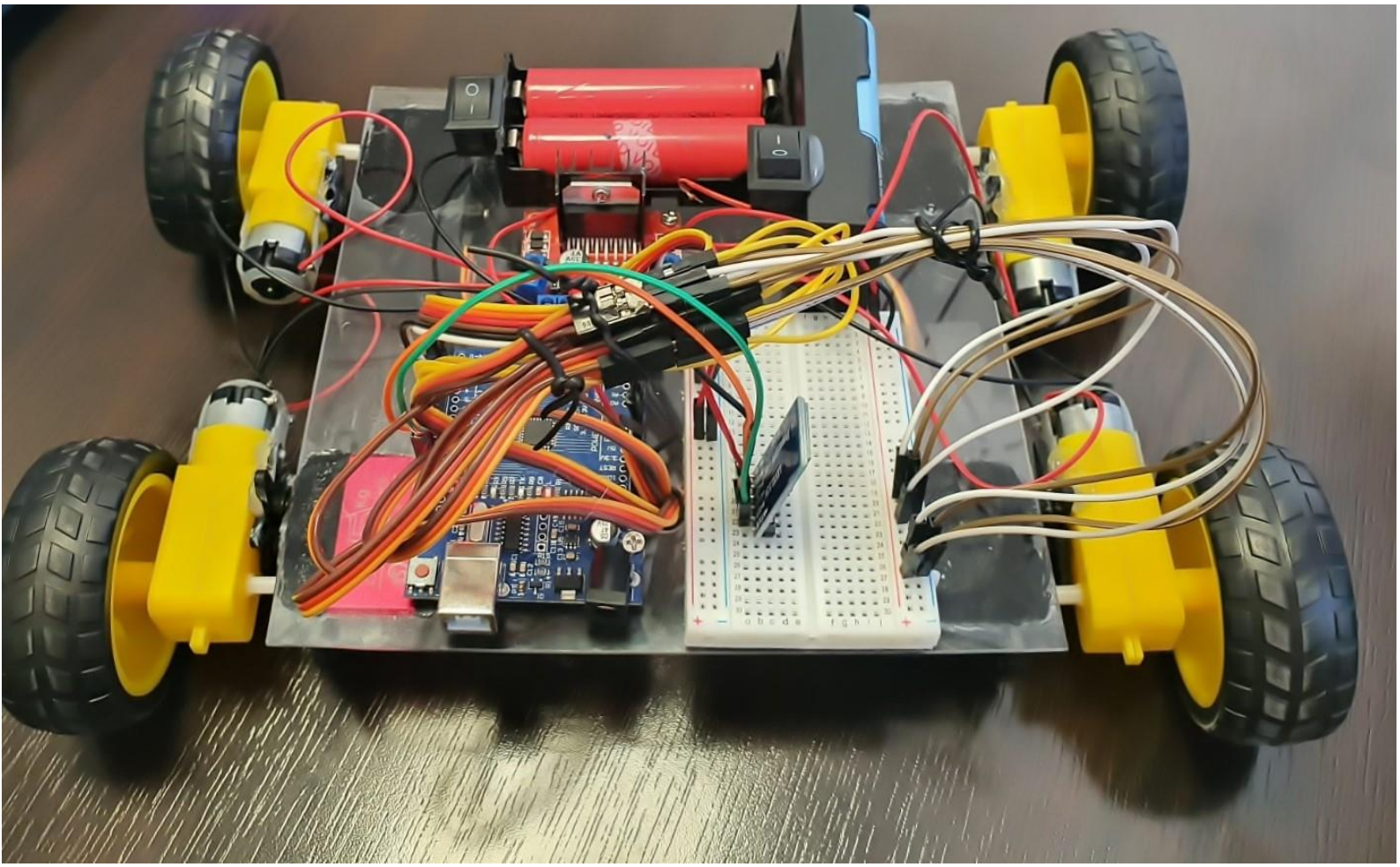
❖ **დამატებითი დეტალები:**

- **PWM სიგნალი** (analogWrite(10, 255)) – აკონტროლებს ძრავას სიჩქარეს (0–255).
- **receiver.resume()** – აღადგენს IR მიმღების მომსახურებას ახალი სიგნალის გამოსაგზავნად.

ეს კოდი იყენებს IRremote ბიბლიოთეკას IR სიგნალების დასამუშავებლად და აკონტროლებს ძრავებს L298N დრაივერის საშუალებით.

VIII პროექტი

რობოტი მანქანა მოძრავი კორპუსით



საჭირო რესურსი

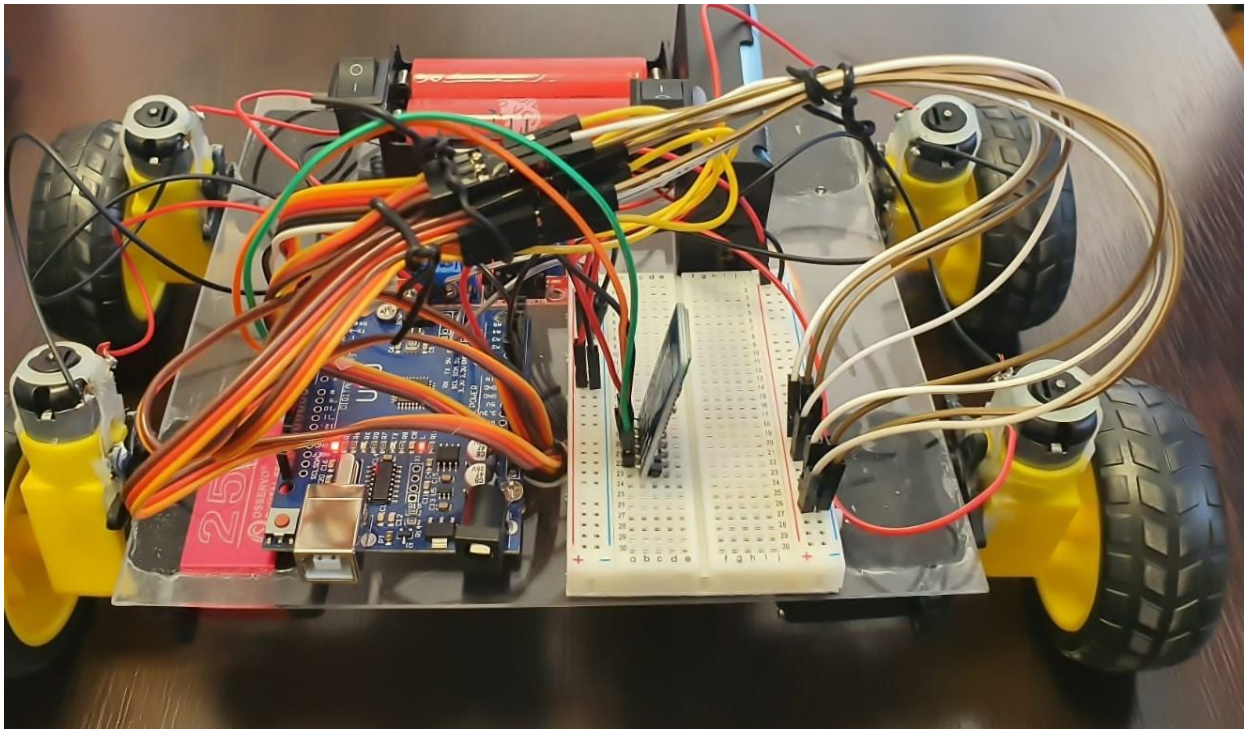
- 4WD მანქანის კომპლექტი
- Arduino UNO
- ძრავას დრაივერი L298N
- Servo ძრავა DS3225MG 25KG x 4
- სამაკეტო დაფა
- გამტარები: მამალ-მამალი
- 4 ცალი ლითიუმის ელემენტი 3.7V და 2 ბუდე
- ჩამრთველ-გამომრთველი x 2
- Bluetooth HC-05 მოდული

საჭირო ბიბლიოთეკა

- SoftwareSerial
- Servo

როგორ მუშაობს რობოჯი მანქანა

ამ პროექტში ჩვენ შევქმნით 4WD მანქანას, რომელიც იმართება ბლუთუზით და დაბრკოლების გადასალახავად, Servo ძრავების საშუალებით, შეუძლია კორპუსის აწევა. კორპუსის ასაწყობად თავად უნდა გამოჭრათ დაფა: 13სმ X 17სმ (იხ. სურათი 1). დაფა სასურველია იყოს 2-3 მმ-იანი ორგმინა ან ფანერა.



სურათი 1. რობოტი მანქანა აწეული კორპუსით (Srevo ძრავების საშუალებით, DC ძრავები ვერტიკალურ მდგომარეობაშია).

თუ თქვენთვის ცნობილია ჩვენ მიერ გამოცემული მეთოდური წიგნი „არდუინოს შემსწავლელი სახელმძღვანელო - 20 STEM პროექტი“ (I-II ნაწილი, 2024 წ.), შეამჩნევდით, რომ ამ პროექტში პირველად ვიყენებთ Servo ძრავის მოდელს DS3225MG 25KG (იხ. სურათი 2). ამ არჩევანის მიზეზი განაპირობა უფრო ძლიერი Servo ძრავების საჭიროებამ, ვიდრე არის მოდელი L298N.

წიგნის ელექტრონული ვერსიები შეგიძლიათ იხილოთ შემდეგ მისამართებზე:

https://chkhirkedela2019.wixsite.com/mysite/about-5

https://educationhouse.ge/page/313

სურათი 2. Servo ძრავა DS3225MG 25KG

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino UNO
ENA	D5
IN1	D4
IN2	D3
IN3	D2
IN4	D7
ENB	D6

2. დააკავშირეთ ერთმანეთთან არდუინო და Servo ძრავების S პინები (GND და +5V შესაბამისად, არდუინოს პინებთან: GND და +5V):

Servo ძრავა	Arduino UNO
წინა მარჯვენა	D8
წინა მარცხენა	D9
უკანა მარჯვენა	D12
უკანა მარცხენა	D13

3. დააკავშირეთ ერთმანეთთან არდუინო და Bluetooth HC-05 მოდული:

Bluetooth HC-05	Arduino Nano
TX	11
RX	10

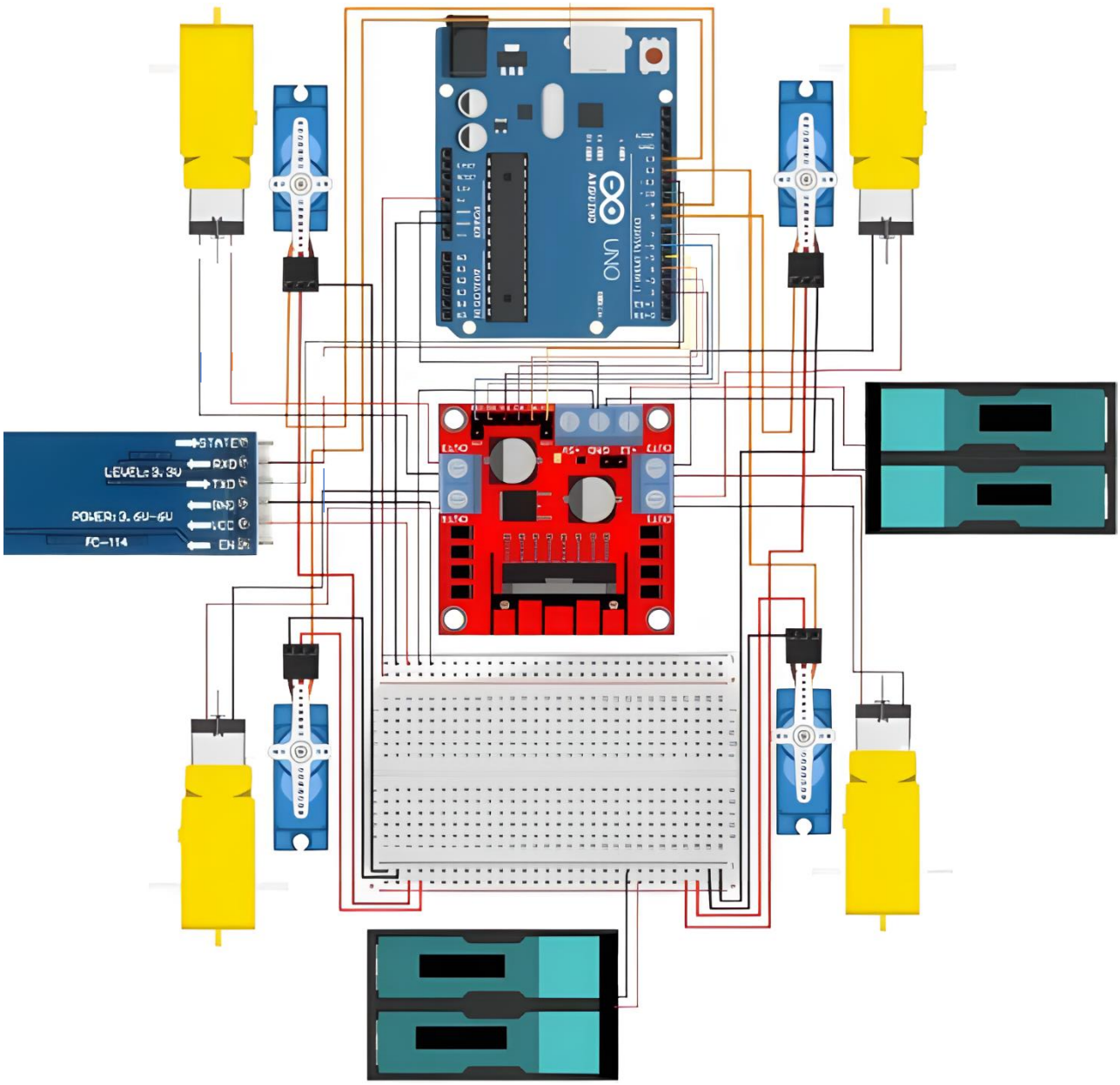
ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

4. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი, Servo ძრავები და ბლუთუზი HC-05 (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino UNO	სამაკეტო დაფა	ძრავას დრაივერი L298N	სამაკეტო დაფა	Servo ძრავები DS3225MG 25KG	სამაკეტო დაფა	ბლუთუზი HC-05	სამაკეტო დაფა
5V	+	+5V	+	წით. გამტარი	+	VCC	+
GND	GND	GND	GND	GND	GND	GND	GND

კიდევ ერთ მნიშვნელოვან დეტალზე გვინდა გავამახვილოთ თქვენი ყურადღება. ვინაიდან ჩვენ ამ პროექტში ვიყენებთ Servo ძრავების განსხვავებულ, უფრო ძლიერ მოდელს, ჩვენი რჩევა იქნება, Servo ძრავების GND და +5V პინები დააკავშიროთ სამაკეტო დაფის GND და +5V -ის მეორე ლიანდაგზე და მისცეთ დამოუკიდებელი კვების წყარო (იხ. სურათი 3. სქემა).



სურათი 3. სქემა

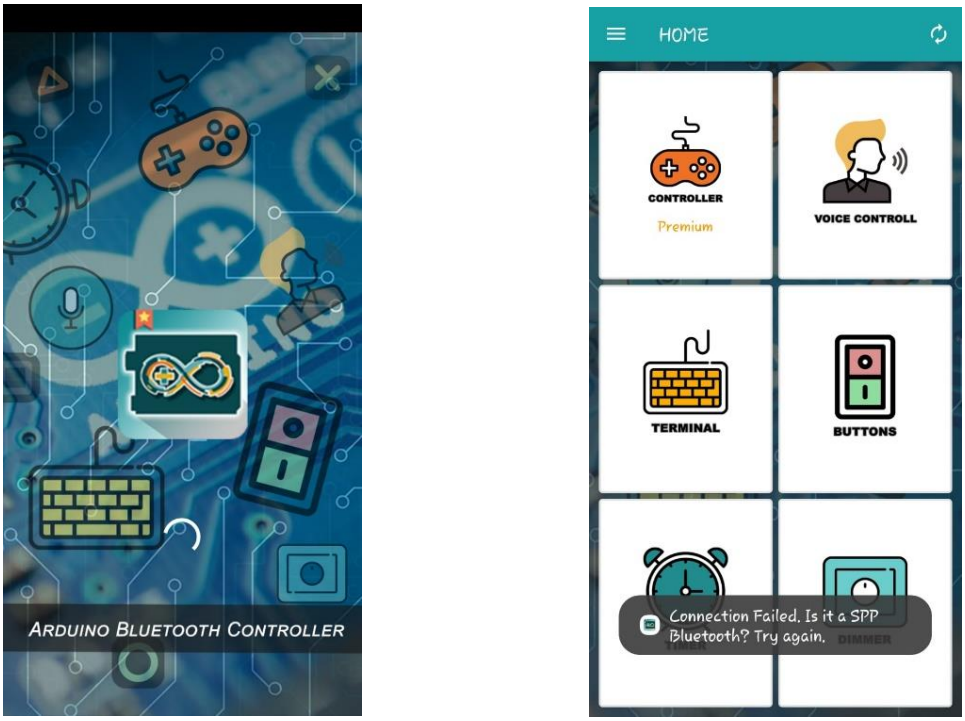
5. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ პორტი.

6. ჩატვირთეთ კოდი.

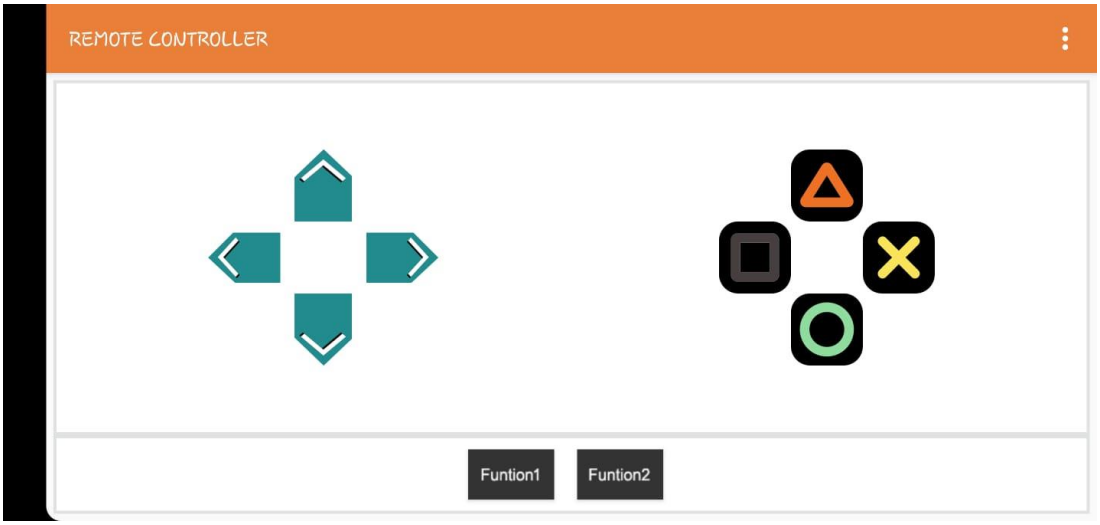
7. ბოლოს, თქვენ მიერ შექმნილი რობოტი მანქანის სამართავად დაგჭირდებათ მობილური ტელეფონის აპლიკაცია, მაგალითად: Arduino Bluetooth Controller ან სხვა ნებისმიერი. შეგიძლიათ თავად შექმნათ აპლიკაცია ან თქვენთვის საჭირო ბრძანებები (წინ, უკან, მარჯვნივ, მარცხნივ, ზემოთ, ქვემოთ, გაჩერება) აპლიკაციის ტერმინალში. ნიმუშისთვის შემოგთავაზებთ ერთ-ერთ ასეთ აპლიკაციას (იხ. სურათი 4):



სურათი 4. აპლიკაცია Arduino Bluetooth Controller და მისი ინტერფეისი

რობოტის მართვა ხდება Controller-იდან, ხოლო ბრძანებების შექმნა - Terminal-დან.

ამ აპლიკაციის Controller-ის ინტერფეისი ასე გამოიყურება (იხ. სურათი 5):



სურათი 5. დისტანციური მართვის ღილაკები

თქვენ ამ მართვის ღილაკებისთვის უნდა შექმნათ ბრძანებები, რომ გარდა წინ და უკან, მარჯვნივ და მარცხნივ სიარულისა, შეძლოთ რობოტი მანქანის კორპუსის აწევა და დაწევა, გაჩერება. ასე გამოიყურება ეს ბრძანებები, რომლებსაც ჩაწერთ ტერმინალში (იხ. სურათი 6):

Forward - > F

Back - > B

Right - > R

Left - > L

Up - > U

Down - > D

STOP - > S

სურათი 6. ტრემინალი

ტერმინალში ბრძანებების ჩაწერის დროს ბლუთუზი ჩართული უნდა იყოს.

კოდის მუშაობის პრინციპი

1. სტრუქტურა და კომპონენტები

- **ბიბლიოთეკები:**
  - SoftwareSerial - Bluetooth კომუნიკაციისთვის
  - Servo - Servo ძრავების კონტროლისთვის
- **კომპონენტები:**
  - 2 DC ძრავა L298D მართვის მოდულით
  - 4 Servo ძრავა
  - Bluetooth მოდული (HC-05 ან HC-06)

2. Bluetooth კომუნიკაცია

კოდი იყენებს SoftwareSerial Bluetooth მოდულს (პინები 10-RX, 11-TX). მოდული იღებს ბრძანებებს და აგზავნის სერიალ მონიტორზე.

3. ბრძანებების სისტემა

ბრძანება	მოქმედება
F	წინ
B	უკან
L	მარცხნივ
R	მარჯვნივ
S	გაჩერება
U	Servo ძრავები ზემოთ (+10°)
D	Servo ძრავები ქვემოთ (-10°)

4. **DC ძრავების კონტროლი**

- **L298D მართვის ლოგიკა:**
  - ENA/ENB - სიჩქარის კონტროლი (PWM)
  - IN1-IN4 - ბრუნვის მიმართულება

5. მიმართულებები:

- **წინ:** IN1=HIGH, IN2=LOW, IN3=HIGH, IN4=LOW
- **უკან:** IN1=LOW, IN2=HIGH, IN3=LOW, IN4=HIGH
- **მარცხნივ:** IN1=HIGH, IN2=LOW, IN3=LOW, IN4=HIGH
- **მარჯვნივ:** IN1=LOW, IN2=HIGH, IN3=HIGH, IN4=LOW

6. Servo ძრავების კონტროლი

❖ 4 Servo განლაგებულია რობოტის 4 კუთხეში.

- setServos() ფუნქცია:
  - წინა მარჯვენა: `180 - angle` (სარკისებური)
  - წინა მარცხენა: `angle`
  - უკანა მარჯვენა: `angle`
  - უკანა მარცხენა: `180 - angle` (სარკისებური)

ეს უზრუნველყოფს სიმეტრიულ მოძრაობას.

7. **მუშაობის ციკლი**

❖ `setup()` - ინიციალიზაცია

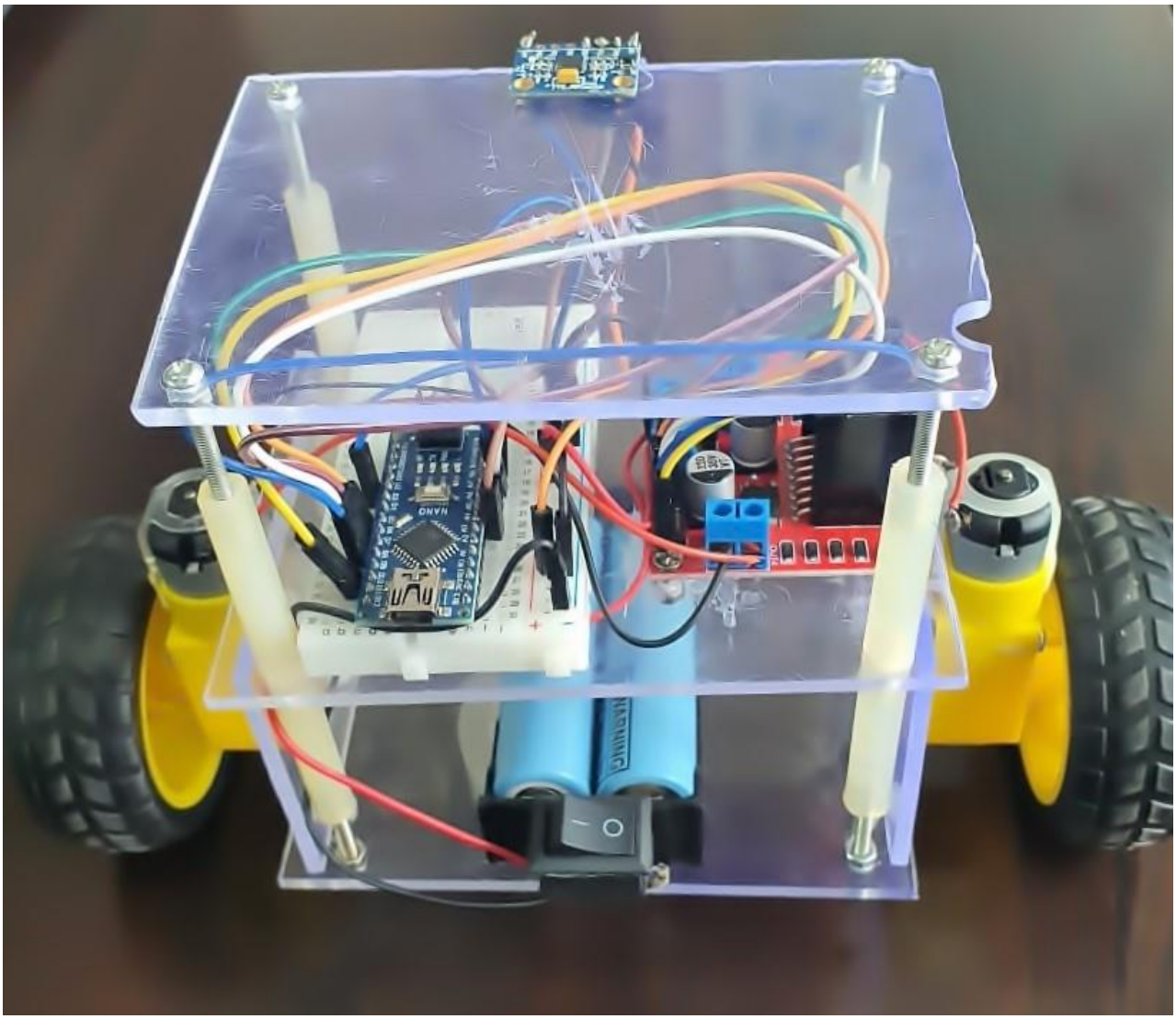
❖ `loop()` - Bluetooth მონიტორინგი

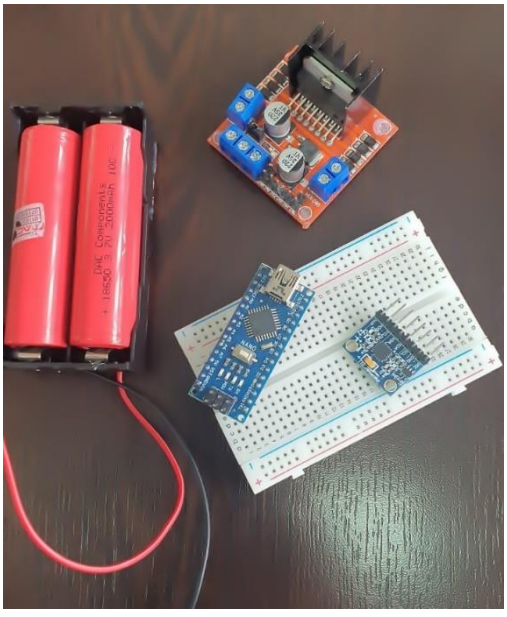
❖ ბრძანების მიღება → შესაბამისი ფუნქციის გამოძახება

❖ პროცესის გამეორება

ეს კოდი შესაფერისია 4-ძრავიანი რობოტისთვის Servo მექანიზმით, რომელიც მართავს Bluetooth-ის მეშვეობით, მობილური აპლიკაციიდან ან კომპიუტერიდან.

თვითბალანსირებადი რობოტი მანქანა





საჭირო რესურსი

- Arduino NANO
- ძრავას დრაივერი L298N
- GY-521 MPU6050 აჩქარების/გიროსკოპის სენსორი
- 2-2 ცალი DC ძრავა და საბურავი
- მამალ-მამალი და დედალ-მამალი გამტარები
- 2 ცალი ლითიუმის ელემენტი 3.7V და ბუდე

საჭირო ბიბლიოთეკა

- MPU6050
- 12Cdev
- PID

თვითბალანსირებადი რობოტი მანქანის მნიშვნელობა

გულისხმობს სისტემის უნარს, ავტომატურად შეინარჩუნოს ვერტიკალური ბალანსი გარე დარღვევების (დახრილობა, დარტყმები, უსწორმასწორო ზედაპირი) მიუხედავად.

- **ბალანსის ხელოვნება:** ჩვენ განვიხილავთ, თუ როგორ გადააქცევს **GY-521 MPU6050** აჩქარების/გიროსკოპის სენსორი ჩვეულებრივ 2WD მანქანას თვითბალანსირებად სისტემად, როგორ იმეორებს/ასრულებს სეგვეის მუშაობის პრინციპს – ეს არის ერთ-ერთი ყველაზე დამაჯერებელი დემონსტრაცია სენსორებისა და ალგორითმების ძალისა.

1. რა არის სეგვეი (SEGWAY)?

სეგვეი არის პირადი ელექტროტრანსპორტის ერთ-ერთი ყველაზე ცნობილი ნიმუში. ეს არის ორბორბალიანი სადგომი, რომელზეც მგზავრი დგას და რომელიც თავისთავად ინარჩუნებს ვერტიკალურ ბალანსს. მისი მართვა ხდება სხეულის დახრილობის მიხედვით: წინ დახრისას ის მიდის წინ, უკან დახრისას – უკან, ხოლო როცა პირდაპირ დგახართ, ის უმოძრაოდ დგას.

2. როგორ მუშაობს სეგვეის პრინციპი?

მთელი მაგია ორ ძირითად კომპონენტშია დამალული:

1. **გიროსკოპი და აჩქარების სენსორი (სწორედ ის, რაც ჩვენს GY-521 MPU6050 მოდულში გვაქვს):** ეს სენსორი მუდმივად "აგრძელებს" თავის პოზიციას სივრცეში. ის ზომავს დახრილობის კუთხეს (რამდენად არის დახრილი მანქანა წინ ან უკან) და ამ დახრილობის კუთხოვან სიჩქარეს.
2. **ჭკვიანი კონტროლერი (Arduino ან მსგავსი მიკროკონტროლერი):** კონტროლერი წამში ასჯერ ან უფრო ხშირად კითხულობს სენსორის მონაცემებს. როგორც კი აღიქვამს, რომ მანქანა იწყებს დაცემას წინ (მაგალითად, გრავიტაციის გამო), ის მყისიერად უგზავნის ბრძანებას ძრავებს, რომ იმოძრაონ წინ იმ მიმართულებით, საითაც ის დაცემა ხდება, რათა "დაიჭიროს" იგი.

ეს არის ინვერსიული ქანქარის პრინციპი. წარმოიდგინეთ ფანქარი, რომელიც ვერტიკალურად გიდევთ ხელისგულზე და ცდილობთ შეუნარჩონოთ ბალანსი, რომ არ დაეცეს. როდესაც ფანქარი იწყებს წინ დაცემას, თქვენი ხელი მოძრაობს წინ, რათა დაცემა შეაჩეროთ. სეგვეი (და თქვენი რობოტი მანქანა) აკეთებს ზუსტად ამას, მაგრამ ძალიან სწრაფად და ზუსტად.

3. როგორ იმეორებს ამას თქვენი 2WD რობოტი მანქანა?

თქვენი პროექტი, სადაც 2WD მანქანა ხდება თვითბალანსირებადი, არის სეგვეის პრინციპის პირდაპირი და ელეგანტური იმიტაცია.

- **სენსორი:** თქვენი GY-521 MPU6050 მოდული ასრულებს იგივე როლს, რასაც სეგვეის გიროსკოპული სენსორების სისტემა. ის აწვდის მონაცემებს მანქანის კუთხის და აჩქარების შესახებ.
- **კონტროლერი:** თქვენი მიკროკონტროლერი გაშვებულია სპეციალური ალგორითმით (ხშირად PID კონტროლერი), რომელიც ამ მონაცემების მიხედვით ითვლის, რა სიჩქარე და მიმართულება სჭირდება ძრავებს ბალანსის შესანარჩუნებლად.
- **ძრავები:** L298N ძრავას დრაივერის მეშვეობით, კონტროლერი მართავს თქვენი ორი ძრავას სიჩქარეს. თუ მანქანა იწყებს წინ ან უკან დაცემას, ძრავები მოძრაობენ შესაბამისი მიმართულებით, სიჩქარით, რომელიც PID კონტროლერის მიერ გამოთვლილი პროპორციული რეაგირების შედეგია.

თქვენი რობოტი მანქანა იყენებს იმავე ფუნდამენტური ფიზიკის კანონებსა და კონტროლის სტრატეგიას, რასაც კომერციული სეგვეი, რათა დაიჭიროს საკუთარი თავი და დადგეს პირდაპირ, ზუსტად ისე, როგორც ამას აკეთებს ინვერსიული ქანქარა.

ეს არის ერთ-ერთი ყველაზე შთამბეჭდავი დემონსტრაცია რობოტოტექნიკაში, რადგან ის ნათლად აჩვენებს, თუ როგორ შეუძლია "მარტივ" სენსორს და "ჭკვიან" პროგრამას ერთად შექმნან კომპლექსური და დინამიური ქცევა.

რატომ არის ეს მნიშვნელოვანი?

თვითბალანსირებადი სისტემები აერთიანებს:

✅ მექანიკას (ძრავები, სენსორები),

✅ ელექტრონიკას (მიკროკონტროლერები),

✅ პროგრამულ უზრუნველყოფას (ალგორითმები).

ეს პროექტი არის მიკროსკოპული მოდელი თანამედროვე ავტომატიზირებული სისტემებისა, რომლებითაც გარშემორტყმულნი ვართ (მაგ., Tesla-ს ავტოპილოტი, ჭკვიანი სათამაშოები) და მოიცავს:

1. შებრუნებული ქანქარის მოდელი

შებრუნებული ქანქარის მოდელი ილუსტრირებს დინამიურ სტაბილიზაციას, რომელიც გამოიყენება რობოტოტექნიკასა და ავტომატური მართვის სისტემებში.

PID კონტროლის ვიზუალიზაცია: საუკეთესო საშუალებაა კონტროლის თეორიის სწავლებისთვის.

2. პრაქტიკული გამოყენება

სეგვეები და პერსონალური ტრანსპორტი: ბალანსირების ალგორითმები გამოიყენება ელექტრო სკუტერებში, ჰოვერბორდებში და ელექტრო ურიკებში.

სამრეწველო რობოტები: მაგ., საწყობების ავტომატიზირებული სისტემები, რომლებიც აწონასწორებენ ტვირთს.

კოსმოსური ტექნოლოგიები: თვითბალანსირებადი სისტემები
გამოიყენება სატელიტების და როვერების სტაბილიზაციაში.

3. საგანმანათლებლო მნიშვნელობა

ინჟინერიის სტუდენტებისთვის: გვიჩვენებს, თუ როგორ
ინტეგრირდება სენსორები, ალგორითმები და აქტუატორები რეალურ სისტემაში.

პროგრამირების უნარები: გულისხმობს Arduino/C++-ის გამოყენებას, რეალურ დროში მონაცემთა დამუშავებას.

პრობლემის გადაჭრა: სწავლობს კალიბრაციის, ხმაურის ფილტრაციის და სისტემის ოპტიმიზაციის მეთოდებს.

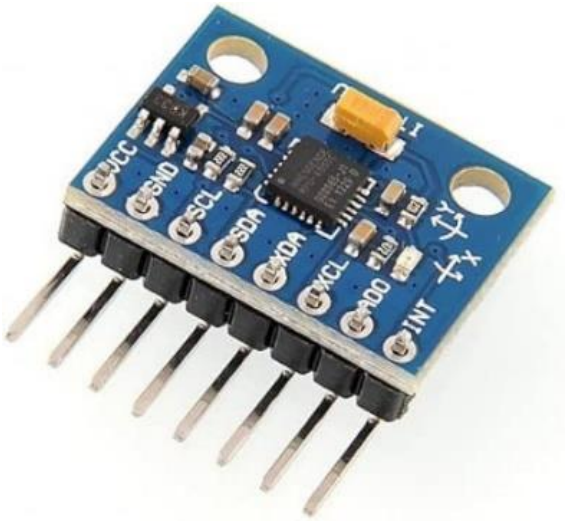
4. ტექნოლოგიური ინოვაციების საფუძველი

ავტონომური რობოტები: ბალანსირების უნარი აუცილებელია ორფეხა
რობოტებისთვის ან დრონებისთვის.

5. მედიცინა: რეაბილიტაციური მოწყობილობები (ექსოსკელეტონები) იყენებენ მსგავს პრინციპებს.

ეს პროექტი განსხვავებულია ჩვენ მიერ უკვე შემოთავაზებული რობოტი მანქანებისგან. მასზე მუშაობის დროს თქვენ დაგჭირდებათ გაცილებით მეტი დრო და მოთმინება.

პროექტში ვიყენებთ GY-521 MPU6050 აჩქარების/გიროსკოპის სენსორს (იხ. სურათი 1), რომელიც ზომავს დახრილობის კუთხეს (Pitch) და კუთხურ სიჩქარეს (Gyro).



სურათი 1. GY-521 MPU6050 აჩქარების/გიროსკოპის სენსორი

ავაწყოთ სქემა

ვიდრე 2WD თვითბალანსირებადი რობოტი მანქანის მოქმედების პრინციპების განხილვაზე გადავიდოდეთ, ავაწყოთ სქემა (იხ. სურათი 2).

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino NANO
ENA	Pin D9
IN1	Pin D5
IN2	Pin D6
IN3	Pin D7
IN4	Pin D8
ENB	Pin D10

2. დააკავშირეთ ერთმანეთთან არდუინო და GY-521 MPU6050 აჩქარების/გიროსკოპის სენსორი (ვიყენებთ სენსორის მხოლოდ 5 პინს):

GY-521 MPU6050	Arduino NANO
VCC (+5V)	+5V
GND	GND
SCL	Pin A5
SDA	Pin A4
INT	Pin D2

ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

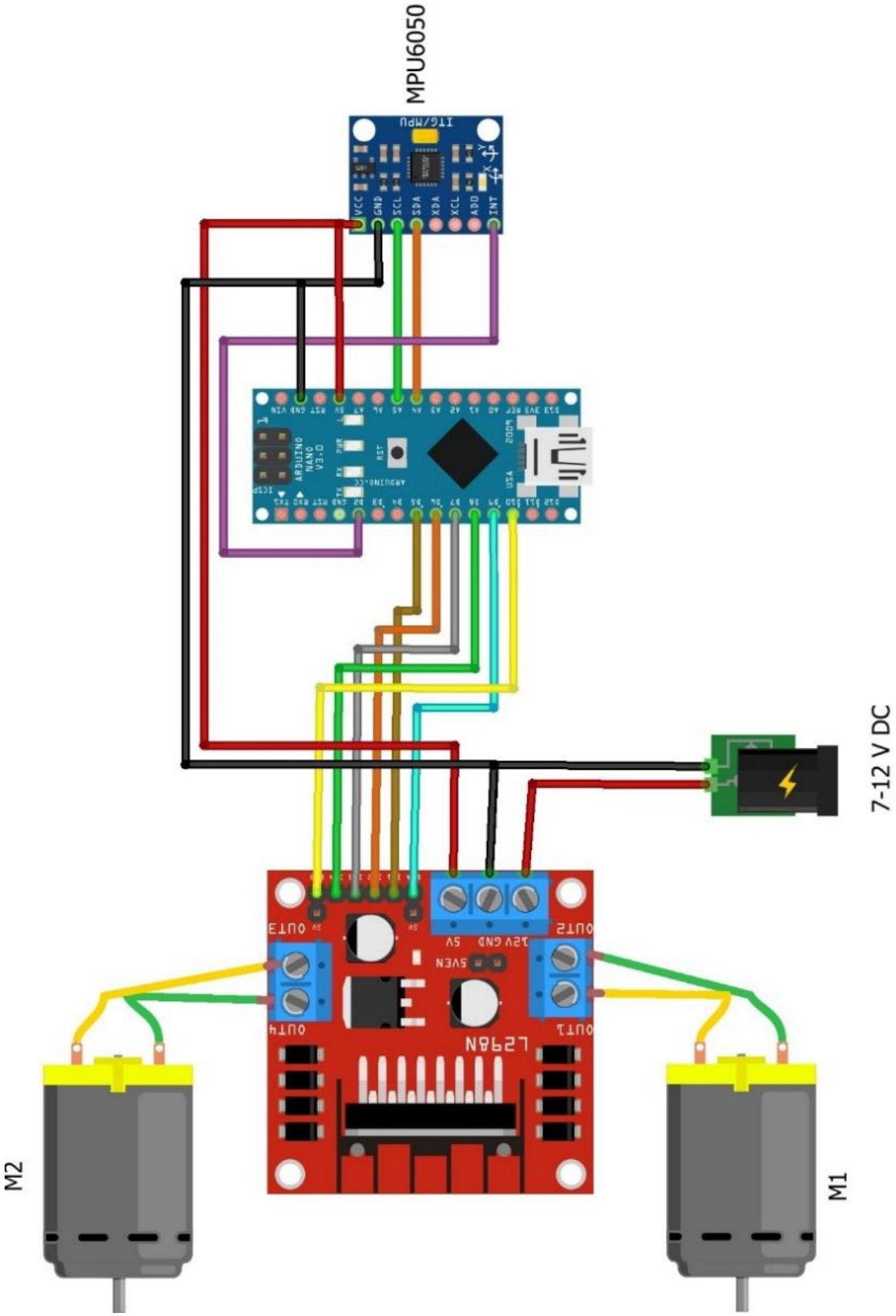
3. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი და GY-521 MPU6050 აჩქარების/გიროსკოპის სენსორი (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino NANO	სამაკეტო დაფა
5V	+
GND	GND

ძრავას დრაივერი L298N	სამაკეტო დაფა
+5V	+
GND	GND

GY-521 MPU6050	სამაკეტო დაფა
VCC	+
GND	GND

პროექტის სქემა ასე გამოიყურება (იხ. სურათი 2):



სურათი 2. სქემა

4. მას შემდეგ რაც ყველა სენსორს ერთმანეთთან დავაკავშირებთ, ძირითადი კოდის ჩატვირთვამდე საჭიროა დავაკალიბროთ GY-521 MPU6050 აჩქარების/გიროსკოპი. ამისათვის ჩავტვირთოთ კოდი MPU6050-Calibration.

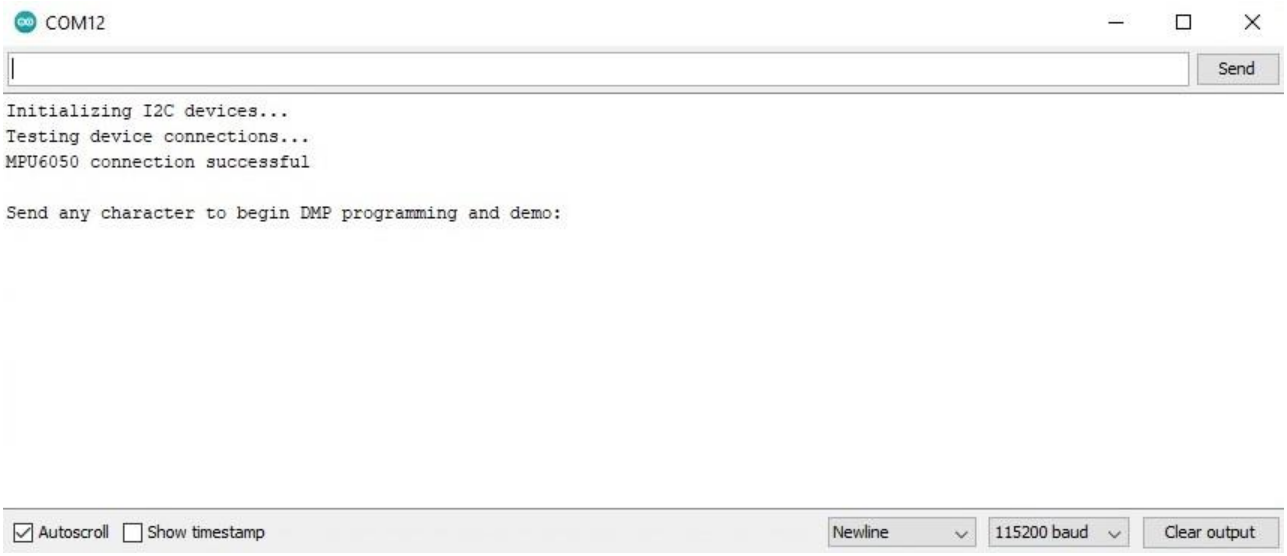
5. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ Arduino NANO და პორტი.

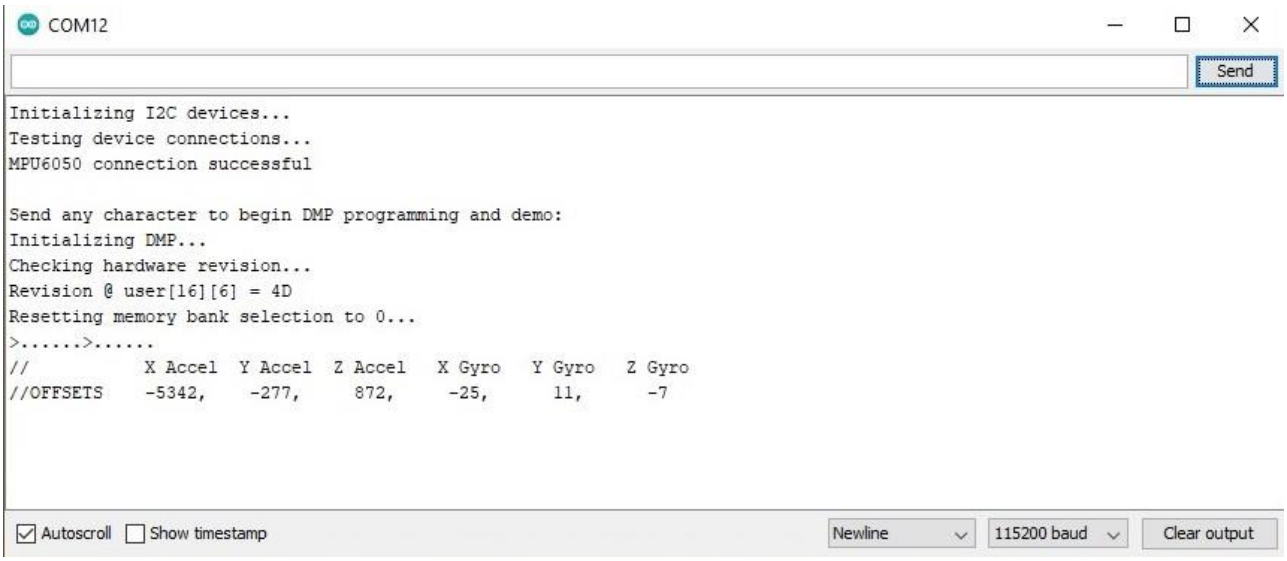
6. ჩატვირთეთ კოდი.

კოდის ჩატვირთვის შემდგომ, ისე რომ არ ვთიშავთ არდუინოს IDE ინტერფეისს, აქვე ვხსნით მიმდევრობითი კომუნიკაციის მონიტორს (Serial Monitor), სადაც უნდა გამოვიდეს მესამე სურათზე გამოსახული ტექსტი, რაც ნიშნავს, რომ სენსორთან კავშირი წარმატებულია (იხ. სურათი 3ა).



სურათი 3 ა

7. შემდეგ, ამავე ფანჯრის საძიებო ველში ვწერთ, მაგალითად, 1-ს (როგორც ამას მოითხოვს ფანჯარაში გამოტანილი შეტყობინება) და ვაგზავნით. ამჯერად ჩვენ უნდა მივიღოთ MPU6050 გიროსკოპის კალიბრაციის მნიშვნელობები (იხ. სურათი 3 ბ). ეს მნიშვნელობები ყველასთვის განსხვავებულია, რაც დამოკიდებულია სხვადასხვა ფაქტორზე: მანქანის სიდიდე, სიმძიმე, სენსორის განთავსების ადგილი...



სურათი 3 ბ

8. მიღებული მნიშვნელობები უნდა ამოიწეროთ და ძირითადი კოდის ჩატვირთვამდე, უკვე არსებული მნიშვნელობები ჩაანაცვლოთ თქვენი მნიშვნელობებით (იხ. სურათი 4).

სურათი 4.

9. ჩატვირთეთ კოდი.

ყველა ზემოთ ჩამოთვლილი ნაბიჯის ზედმიწევნით შესრულების შემდეგ ვიწყებთ პროცესს, რომლის საშუალებითაც თქვენმა მანქანამ უნდა შეძლოს თვითბალანსირება - PID კოეფიციენტების მორგება.

გახსოვდეთ, რომ თვითბალანსირებადი რობოტის კალიბრაციას დრო და მოთმინება სჭირდება. დაიწყეთ პატარა მნიშვნელობებით და თანდათან გაზარდეთ ისინი.

PID კოეფიციენტების მორგება/რეგულირება

კოდში მოცემულია PID პარამეტრები:

```
double Kp = 30;
double Kd = 1.5;
double Ki = 100;
```

PID კოეფიციენტების მორგების რეკომენდაციები

დაწყება მხოლოდ P (Kp)-ით (Ki და Kd = 0)

1. გაზარდეთ Kp მანამ, სანამ რობოტი არ დაიწყებს რხევას და ოდნავ მაინც არ დაიჭერს ბალანსს (დაიწყეთ 20-ით და გაზარდეთ თანდათანობით - 10 ერთეულით);
2. დაამატეთ D (Kd) რხევების შესამცირებლად (დაიწყეთ 0.5-ით და გაზარდეთ თანდათანობით - 0.2, 0.3 ან 0.5 ერთეულით);
3. დაამატეთ I (Ki) მცირე ცდომილების აღმოსაფხვრელად (დაიწყეთ 50-ით და გაზარდეთ თანდათანობით - 50 ერთეულით).

დაიწყეთ საწყისი მნიშვნელობებით:

```
double Kp = 20;
double Kd = 0.5;
double Ki = 100;
```

ტესტირების პროცედურა:

- **დააჭირეთ რობოტს ხელი** და დააკვირდით როგორ ცდილობს ბალანსის აღდგენას:
    - თუ **ნელა რეაგირებს**: გაზარდეთ `Kp`.
    - თუ **რხევადია**: გაზარდეთ `Kd`.
    - თუ **მუდმივად იხრება**: გაზარდეთ `Ki`.

კოდის PID პარამეტრებში გვაქვს კიდევ ერთი მნიშვნელობა - **double originalSetpoint = 178.0;**

იდეალურ შემთხვევაში ის უნდა იყოს 180. ჩვენ შემთხვევაში ეს 178.0-ია. მისი ცვლილება უნდა მოხდეს სულ ბოლოს, როცა თქვენი მანქანა მეტ-ნაკლებად შეძლებს თვთბალანსირებას.

- **Setpoint = 180.0** ნიშნავს:
    - რობოტი იდეალურად დაბალანსებულია, როცა **სრულად ვერტიკალურად** დგას
    - ეს არის **"ნულოვანი წერტილი"** თქვენი PID კონტროლერისთვის

პრაქტიკული მაგალითი:

- **input = 180.0** → რობოტი სრულად ვერტიკალურია ✅
- **input = 170.0** → რობოტი ოდნავ წინ იხრება (ძრავებმა უკან უნდა იმოძრაონ)
- **input = 190.0** → რობოტი ოდნავ უკან იხრება (ძრავებმა წინ უნდა იმოძრაონ)

რჩევა:
თუ რობოტი **სტაბილურად ვერ იკავებს ბალანსს**, შეგიძლიათ ოდნავ შეცვალოთ ეს მნიშვნელობა:

```
double originalSetpoint = 182.0;  // ოდნავ წინ იხრება
// ან
double originalSetpoint = 178.0;  // ოდნავ უკან იხრება
```

ეს შეიძლება დაგჭირდეთ, თუ რობოტის **ცენტრის სიმძიმე ოდნავ გადანაცვლებულია**.

გახსოვდეთ, რომ კალიბრაციას დრო დასჭირდეს. მოთმინება და თანდათანობითი ცვლილებებია საჭირო.

2WD თვითბალანსირებადი რობოტი მანქანის მოქმედების პრინციპები

მარტივად რომ ავხსნათ: მანქანა ინარჩუნებს ვერტიკალურ ბალანსს **მოტორების სიჩქარის რეგულირებით**, რომელიც დაფუძნებულია დახრილობის კუთხის სენსორებით გაზომვაზე.

❖ **ძირითადი კომპონენტები**

- **MPU6050 (IMU სენსორი)** – ზომავს **დახრილობის კუთხეს** (Pitch) და **კუთხურ სიჩქარეს** (Gyro).
- **PID კონტროლერი** – ადგენს მოტორების სიჩქარეს (output) დახრილობის მიხედვით.
- **DC ძრავები** (2 ცალი) – აბალანსებს მანქანას წინ/უკან მოძრაობით.
- **Motor Driver (L298N ან მსგავსი)** – აკონტროლებს ძრავების სიჩქარესა და მიმართულებას.

❖ **როგორ მუშაობს?**

1. **სენსორი აფიქსირებს დახრილობას:**
   - MPU6050 ამოიცნობს, რამდენი გრადუსითაა დახრილი მანქანა (Pitch).
   - თუ მანქანა წინ იხრება, სენსორი ამას აფიქსირებს და აგზავნის მონაცემებს Arduino-ზე.

2. **PID კონტროლერი ითვლის "შეცდომას":**
   - input = ამჟამინდელი დახრილობა (მაგ., 178°).
   - setpoint = სასურველი დახრილობა (ჩვეულებრივ 175.8° ~ ვერტიკალი).
   - error = სხვაობა setpoint-სა და input-ს შორის.
   - output = PID გამოთვლის სიგნალს ძრავებისთვის (Kp * error + Kd * დახრილობის სიჩქარე).

3. **ძრავები აბალანსებენ მანქანას:**
   - თუ მანქანა წინ იხრება, ძრავები უკან მოძრაობენ (output > 0).
   - თუ უკან იხრება, ძრავები წინ მიდიან (output < 0).
   - PID ავტომატურად არეგულირებს სიჩქარეს, რომ input დარჩეს setpoint-თან ახლოს.

- **რატომ არის PID აუცილებელი?** - **PID** არის თქვენი ტვინი, რომელიც განსაზღვრავს, რამდენად სწრაფად და რამდენად ძლიერად უნდა იმოქმედოთ.

    - P (Proportional) – რაც უფრო დიდია დახრილობა, მით უფრო ძლიერია ძრავების რეაქცია.
    - D (Derivative) – აკონტროლებს რხევებს (როგორც "დამამშვიდებელი").
    - I (Integral) – აღმოფხვრის მუდმივ რხევას.

- **რა მოხდება, თუ რომელიმე კომპონენტი არ მუშაობს?**

    - MPU6050 არასწორია → მანქანა ვერ ამოიცნობს დახრილობას → დაეცემა.
    - PID კოეფიციენტები არასწორია → ან არარეაგირება, ან გადაჭარბებული რხევები.
    - ძრავები არ არის დაბალანსებული → მანქანა გადაიხრება ერთი მიმართულებით.

დასკვნა

მანქანა მუშაობს შებრუნებული ქანქარის პრინციპით:

- აფიქსირებს დახრილობას.
- PID განსაზღვრავს საპასუხო ძალას.
- ძრავები ასწორებენ პოზიციას.

კოდის მუშაობის პრინციპი

ძირითადი იდეა

კოდი აწონასწორებს რობოტს ისევე, როგორც ადამიანი ინარჩუნებს ბალანსს - როდესაც დახრილობა ვლინდება, ძრავებს აძლევს სიგნალს, რობოტი საპირისპირო მიმართულებით დააბრუნონ.

მუშაობის ძირითადი ეტაპები:

1. სენსორიდან მონაცემების მიღება

```cpp
mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
input = ypr[1] * 180 / M_PI + 180;
```

- MPU6050 სენსორი აღრიცხავს დახრილობას (pitch)
- მონაცემები გარდაიქმნება გრადუსებში (-180°-დან +180°-მდე)

2. PID რეგულატორის მუშაობა

```cpp
PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);
```

- **Setpoint (სამიზნე)**: 178° (თითქმის ვერტიკალური მდგომარეობა)
- **Input**: ფაქტიური დახრილობა სენსორიდან
- **Output**: ძრავების სიჩქარე და მიმართულება

3. PID კოეფიციენტები:

- **Kp = 30** - პროპორციული (რეაგირებს დახრილობის სიდიდეზე)
- **Kd = 1.5** - დიფერენციალური (რეაგირებს დახრილობის სიჩქარეზე)
- **Ki = 100** - ინტეგრალური (ასწორებს დროთა განმავლობაში დაგროვილ შეცდომებს)

4. ძრავების მართვა

```
motorController.move(output, MIN_ABS_SPEED);
```

- თუ `output > 0` - რობოტი მოძრაობს წინ დახრილობის გასასწორებლად
- თუ `output < 0` - რობოტი მოძრაობს უკან დახრილობის გასასწორებლად
- `MIN_ABS_SPEED = 15` - მინიმალური სიჩქარე ხახუნის დასაძლევად

რეალურ დროში მუშაობის ციკლი:

1. **მიმდინარე დახრილობის გაზომვა** ← MPU6050

2. **შეცდომის გამოთვლა** ← (სამიზნე - ფაქტობრივი)
3. **PID გამოთვლა** ← რამდენად სწრაფად და საით უნდა იმოძრაონ ძრავებმა
4. **ძრავების გაშვება** ← ბალანსის აღდგენის მიზნით
5. **გამეორება** (10ms-ში ერთხელ)

ბალანსის მაგალითი:

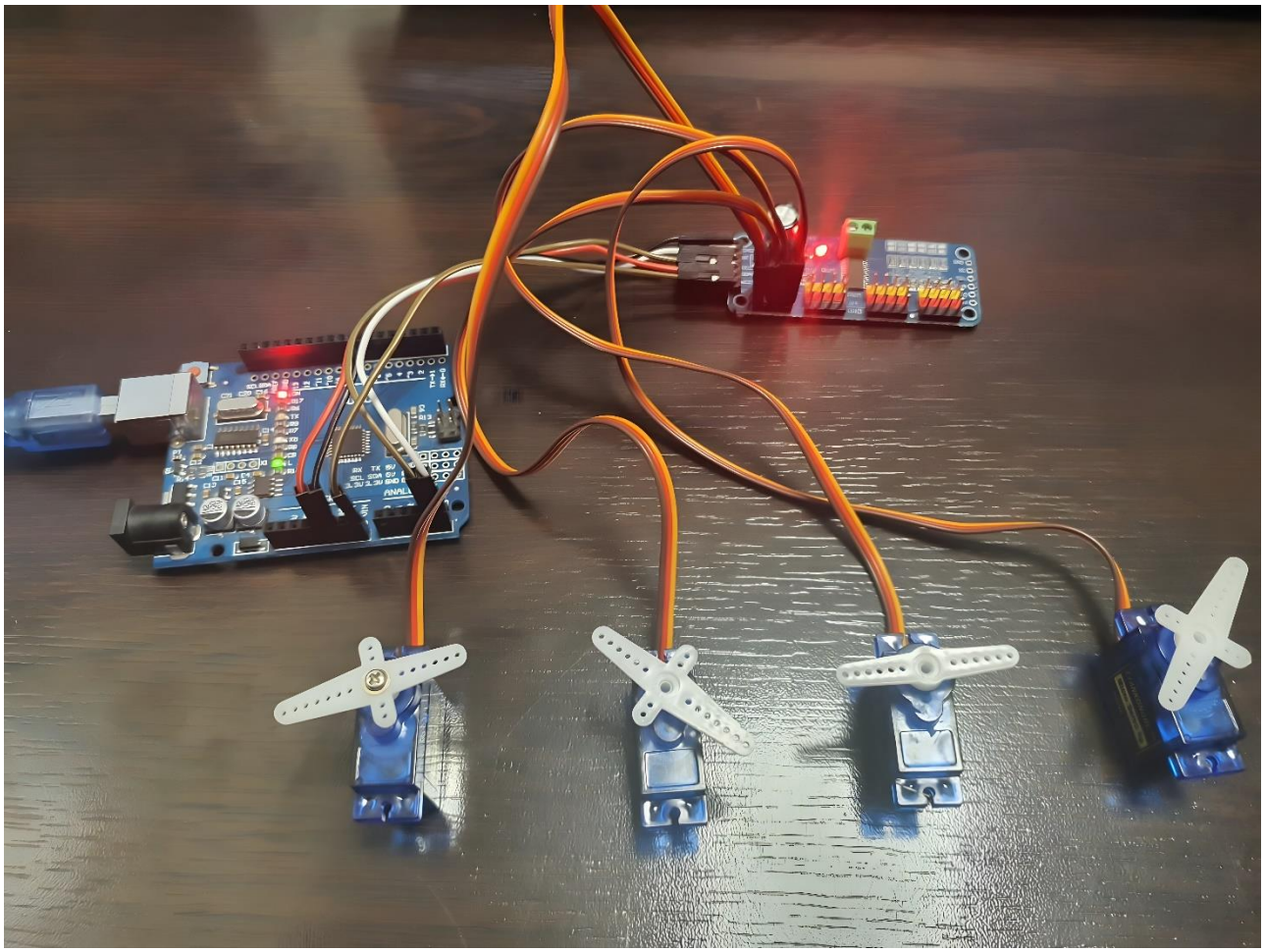
წარმოიდგინეთ, რომ რობოტი იწყებს წინ დახრას:

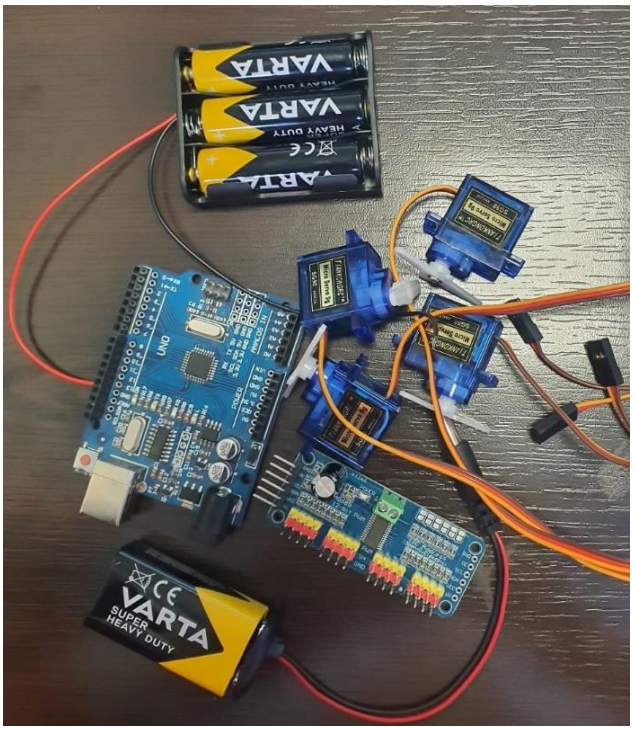
- **სენსორი**: "ვიხრები წინ! (175°)"
- **PID**: "გამოვთვალე - ძრავებმა უნდა იმოძრაონ წინ 45 სიჩქარით"
- **ძრავები**: "მივდივართ წინ - ბალანსი აღდგება!"

ეს პროცესი მიმდინარეობს 100 ჯერ წამში, რაც ქმნის „რბილ“ და სტაბილურ ბალანსირებას!

ეს არის ნამდვილი ინჟინერიის მაგია - სენსორები, ალგორითმები და აქტუატორები ერთად მუშაობენ დინამიური სტაბილურობის შესანარჩუნებლად!

SERVO ძრავების ჭესჭირება და კალიბრაცია





საჭირო რესურსი

- Arduino UNO
- PCA9685 PWM Servo ძრავას დრაივერი (16 არხიანი 12-ბიტიანი)
- 4 ცალი Servo ძრავა Tower Pro SG90 9g
- 3 ცალი 1.5v ბატარეა და ბუდე
- დედალ-მამალი გამტარები
- ელემენტი 9v (კრონას ტიპის)

საჭირო ბიბლიოთეკა

- PWMServoDriver
- Wire

როგორ მუშაობს პროექტი

ამ პროექტში ჩვენ პირველად ვიყენებთ PCA9685 PWM Servo ძრავას მოდულს (იხ. სურათი 1). როგორია ამ მოდულის ფუნქციონალი და მუშაობის პრინციპი:

ძირითადი კონცეფცია

PCA9685 არის 16-არხიანი, 12-ბიტიანი PWM (Pulse Width Modulation) კონტროლერი, რომელიც I^2C ინტერფეისით მართავს Servo ძრავებს, LED-ებს და სხვა PWM-ით მართვად მოწყობილობებს.

1. **PWM (Pulse Width Modulation) პრინციპი**

Servo ძრავასთვის:

პულსის ხანგრძლივობა: 1ms - 2ms (ჩვეულებრივ)

სიხშირე: 50Hz (პერიოდი 20ms)

0 გრადუსი ≈ 1ms პულსი (5% Duty Cycle)

90 გრადუსი ≈ 1.5ms პულსი (7.5% Duty Cycle)

180 გრადუსი ≈ 2ms პულსი (10% Duty Cycle)

2. **PCA9685 ჩიპის არქიტექტურა**

ჩიპის შიგნით:

- 16 PWM გენერატორი (თითოეული არხისთვის)

- 12-ბიტიანი რეზოლუცია (4096 სხვადასხვა მნიშვნელობა)

- I^2C ინტერფეისი

- ინტერნალური საათი

3. **ტექნიკური მხარე**

საათის გამოთვლა:

```
PWM სიხშირე = ოსცილატორის სიხშირე / (4096 × (PRESCALE + 1))

სადაც:

- ოსცილატორი = 25MHz (ტიპიურად)

- PRESCALE = დაყოფის კოეფიციენტი

- 4096 = 12 ბიტის რეზოლუცია (2^12)
```

PRESCALE-ის გამოთვლა Servo ძრავებისთვის (50Hz):

```
PRESCALE = round(25MHz / (4096 × 50Hz)) - 1

        = round(25000000 / (4096 × 50)) - 1

        = round(122.07) - 1

        = 121
```

4. რეესტრების სტრუქტურა

თითოეული არხისთვის 4 რეესტრი:

```
LEDx_ON_L  - ON დროის დაბალი ბაიტი

LEDx_ON_H  - ON დროის მაღალი ბაიტი

LEDx_OFF_L - OFF დროის დაბალი ბაიტი

LEDx_OFF_H - OFF დროის მაღალი ბაიტი
```

5. **I²C კომუნიკაცია**

მისამართი: 0x40 (საწყისი მისამართი)

ადრესირება:

- A0-A5 პინებით შეგიძლიათ შეცვალოთ მისამართი

- მაქსიმუმ 62 მოდული ერთ ბასზე (6 ადრესის პინით)

პრაქტიკული მუშაობის მაგალითი

Arduino-ს კოდიდან მოდულამდე:

// Arduino გზავნის I²C ბრძანებებს:

1. setPWMFreq(50); *// აყენებს სიხშირეს 50Hz-ზე*

2. setPWM(არხი, 0, პულსის_სიგრძე);

// PWM მოდული:

1. იღებს I²C ბრძანებებს SDA/SCL-ზე

2. ინტერპრეტირებს პულსის სიგრძეს (0-4095)

3. გენერირებს შესაბამის PWM სიგნალს

PWM სიგნალის გენერაცია:

ყოველი 20ms პერიოდისთვის (50Hz):

ON დროის დაწყება → დათვლა 0-დან

როცა დათვლა = LEDx_ON → PWM გამოსავალი ხდება HIGH

როცა დათვლა = LEDx_OFF → PWM გამოსავალი ხდება LOW

განმეორება ყოველ 20ms-ში

უპირატესობები

1. **Arduino-სთვის:**

ათავისუფლებს დიგიტალურ პინებს

არ საჭიროებს დამატებით ტაიმერებს

ერთი I²C ბასი მართავს მრავალ მოდულს

2. **PWM ხარისხი:**

12-ბიტიანი რეზოლუცია (4096 სტეპი)

სიზუსტე: 0.25 გრადუსი Servo ძრავებისთვის

სტაბილური სიხშირე

3. **ენერგოეფექტურობა:**

დაბალი ენერგომოხმარება

შეუძლია ექსტერნალური სიმძლავრის მიწოდება

Servo ძრავების მართვის მაგალითი

```
// PWM მნიშვნელობის გამოთვლა Servoსთვის
int pulseWidth(int degrees) {
  // გადაყვანა გრადუსიდან PWM-ში
  float pulse = map(degrees, 0, 180, SERVO_MIN, SERVO_MAX);
  return (int)pulse;
}
// PCA9685-ში:
// პულსის სიგრძე = (OFF_მნიშვნელობა / 4096) × 20ms
```

ტიპიური პარამეტრები Servo ძრავებისთვის

პარამეტრი	მნიშვნელობა	ახსნა
სიხშირე	50Hz	სტანდარტული Servo სიხშირე
პერიოდი	20ms	1/50Hz = 0.02 წამი
PWM რეზოლუცია	12 ბიტი	4096 სხვადასხვა მნიშვნელობა
პულსის სიგრძე	1-2ms	0-180 გრადუსის დიაპაზონი
დაყოფის კოეფიციენტი	121	50Hz-ისთვის

50 ჰერცი გულისხმობს **საკონტროლო სიგნალის სიხშირეს** - Servo ძრავამდე წამში 50-ჯერ იგზავნება სიგნალი.

რეალური მაგალითი მონაცემთა ნაკადით

Arduino: setPWM(0, 0, 307); → I^2C → PCA9685

PCA9685:

1. იღებს მნიშვნელობას 307

2. გადაიყვანს: 307 / 4096 × 20ms ≈ 1.5ms

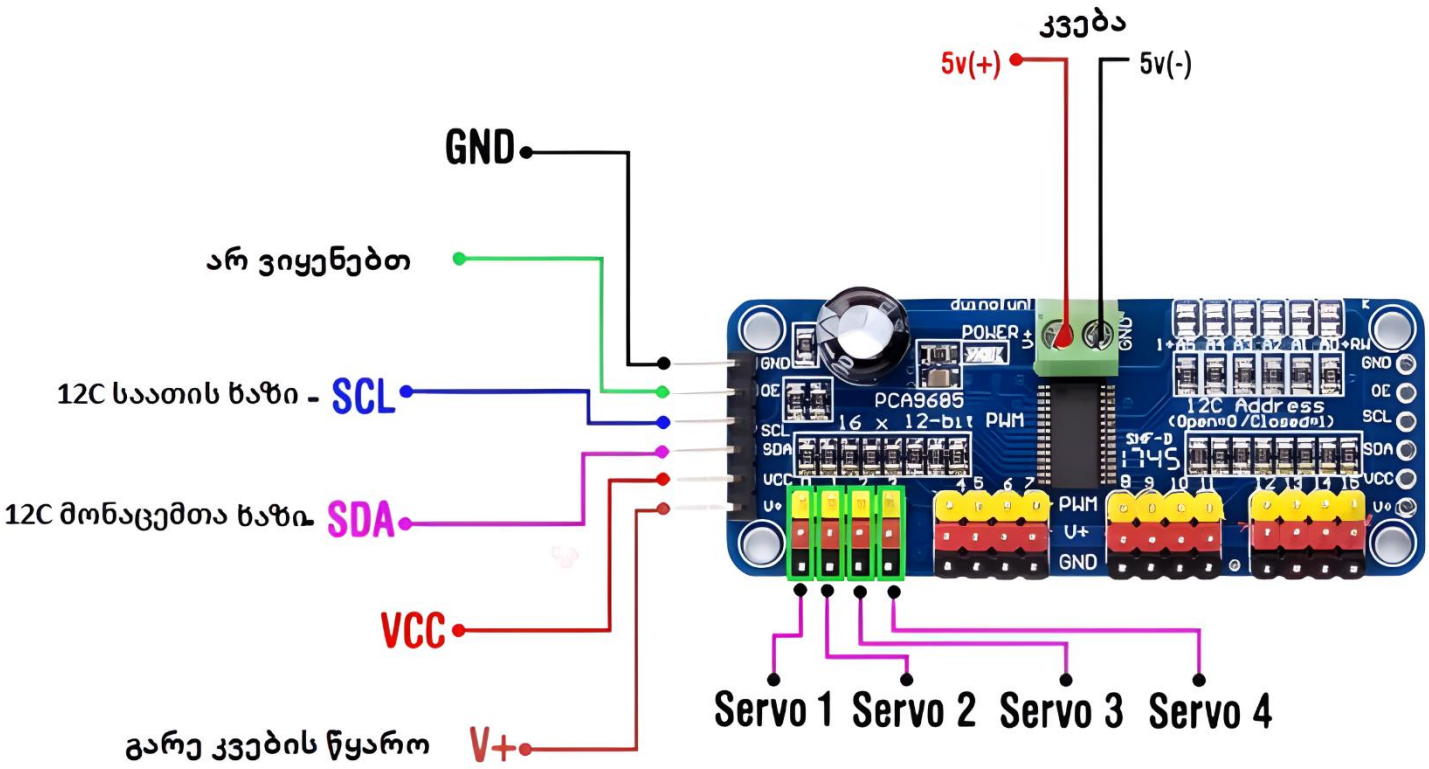
3. გენერირებს PWM სიგნალს:

  - HIGH 0ms-ზე

  - LOW 1.5ms-ზე

  - განმეორება ყოველ 20ms-ში

4. Servo ძრავა ბრუნავს 90 გრადუსამდე



სურათი 1. PCA9685 PWM Servo ძრავას მოდული

დამატებითი ფუნქციები

1. **LED მოდი:** შეუძლია LED-ების მართვაც
2. **საძილე რეჟიმი:** დაბალი ენერგომოხმარების რეჟიმი
3. **სუბ-ადრესი 1-3:** მრავალი მოდულის მართვა
4. **ალტერნატიული ადრესი:** კონფლიქტების თავიდან აცილება

ეს არქიტექტურა საშუალებას აძლევს Arduino-ს ან სხვა მიკროკონტროლერს, მართოს 16 Servo ძრავა მხოლოდ 2 პინის (SDA, SCL) გამოყენებით, რაც მნიშვნელოვნად აფართოებს მის შესაძლებლობებს რობოტიკასა და ავტომატიზაციაში.

ჩვენ შემთხვევაში, PCA9685 PWM Servo ძრავას მოდულით გავტესტავთ როგორც თავად მოდულს, ასევე 4 ცალ Servo ძრავას და გავაკეთებთ მათ კალიბრაციას (იხ. სურათი 2 - სქემა).

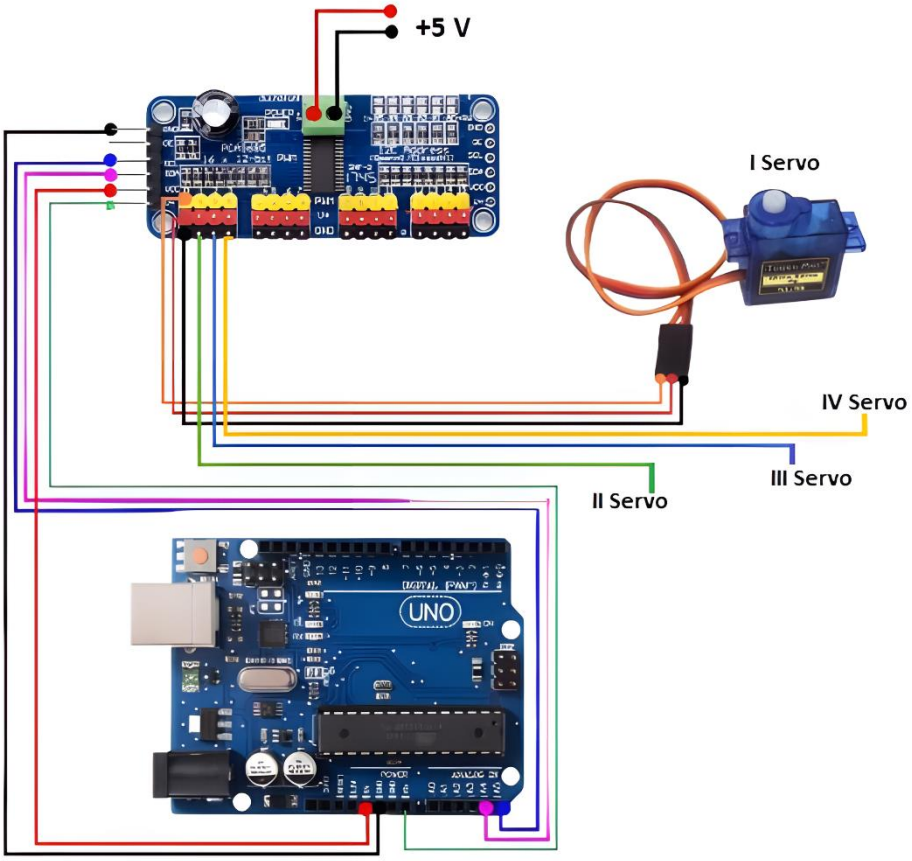
ავაწყოთ სქემა

1. დააკავშირეთ არდუინო და PCA9685 PWM Servo ძრავას მოდული:

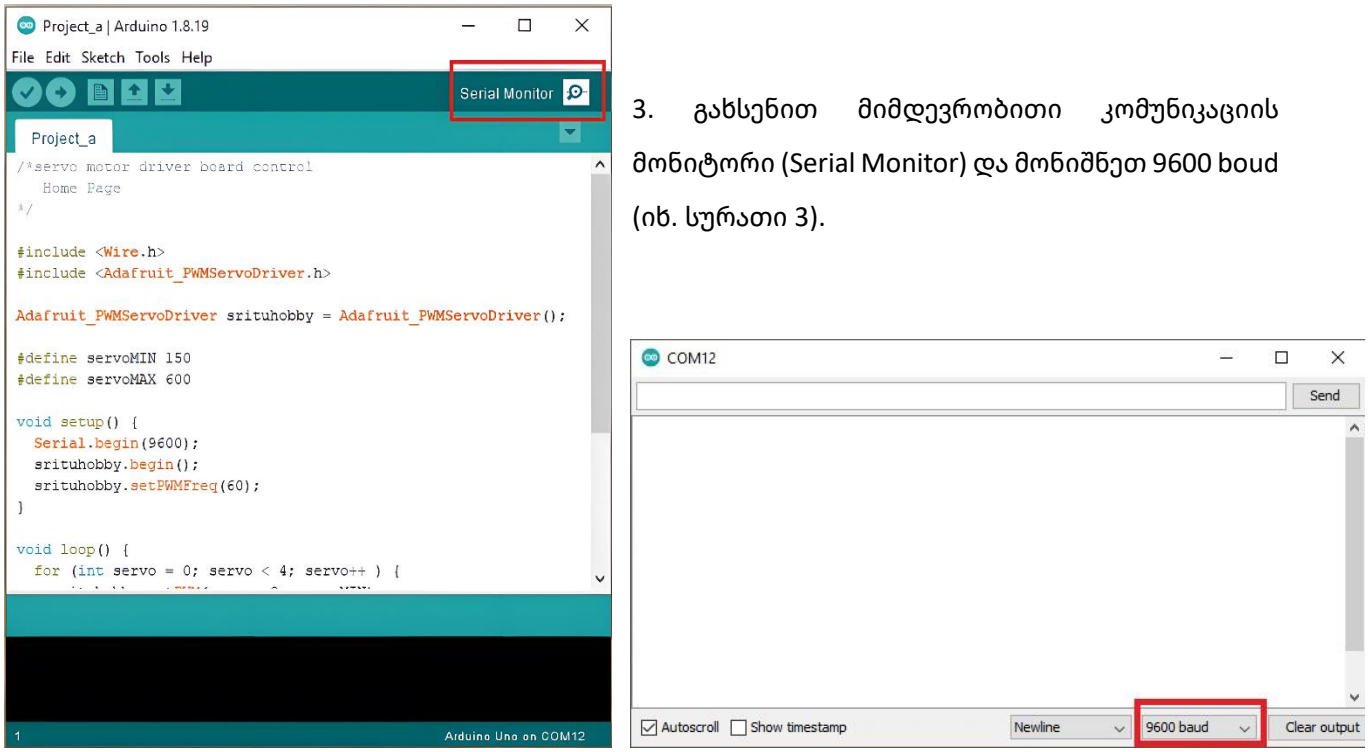
PCA9685 PWM	Arduino UNO
GND	Pin D9
OE	არ ვიყენებთ
SCL	Pin A5
SDA	Pin A4
VCC	+5V
V+ (მხოლოდ საჭიროების შემთხვევაში)	VIN

2. დააკავშირეთ ერთმანეთთან 4 ცალი Servo ძრავა PCA9685 PWM მოდულთან:

PCA9685 PWM მოდულის პორტები	Servo ძრავა
პორტი 0	Servo 1
პორტი 1	Servo 2
პორტი 2	Servo 3
პორტი 3	Servo 4



სურათი 2. სქემა



3. გახსენით მიმდევრობითი კომუნიკაციის მონიტორი (Serial Monitor) და მონიშნეთ 9600 boud (იხ. სურათი 3).

სურათი 3. მიმდევრობითი კომუნიკაციის მონიტორი (Serial Monitor)

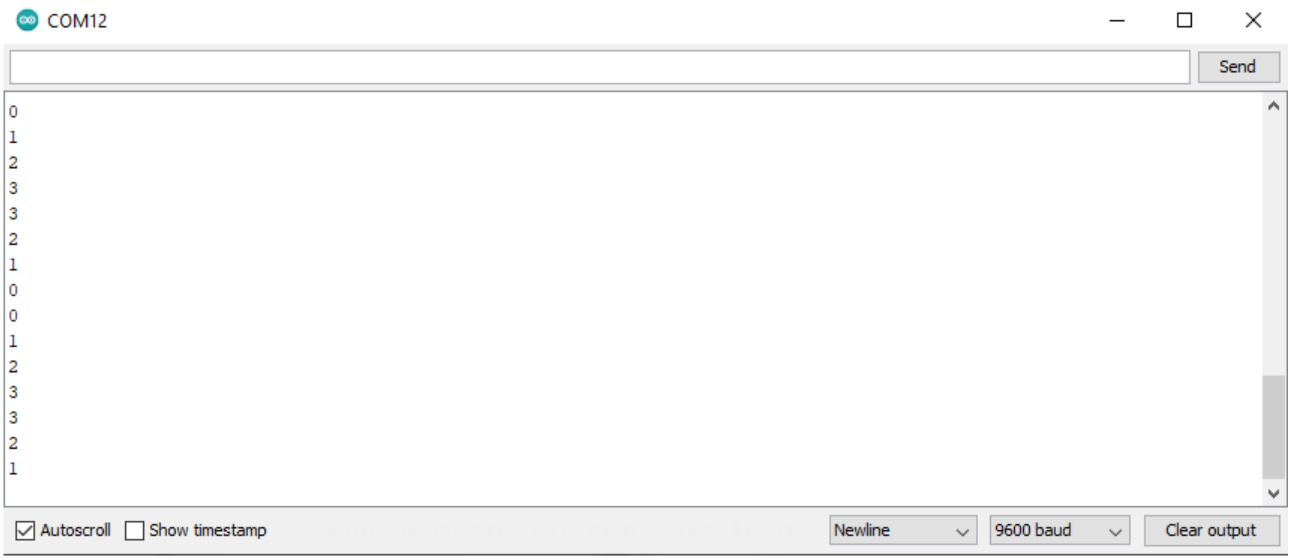
4. დააკავშირეთ არდუინო კომპიუტერთან.

შეხსენება!

კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools და მონიშნეთ Arduino UNO და პორტი.

5. ჩატვირთეთ კოდი (ჩვენ გთავაზობთ ორ კოდს ძრავების შესამოწმებლად - Project_10a და Project_10b. ქვემოთ შემოგთავაზებთ, როგორი მონაცემები უნდა დაგიფიქსირდეთ მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor) თითოეული კოდის შემთხვევაში (იხ. სურათი 4 და სურათი 5).

6. ასეთი მონაცემები უნდა დაფიქსირდეს თქვენს მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor) Project_10a-ს შემთხვევაში (იხ. სურათი 4):



სურათი 4. Project_10a

ეს რიცხვები მიუთითებენ, რომ თქვენი Servo ძრავები დაკავშირებულია PCA9685 PWM Servo ძრავას მოდულის 0, 1, 2, და 3 პორტებთან. თუ სხვა პორტების შემოწმება გაინტერესებთ, ან მეტი Servo გჭირდებათ, მაშინ კოდის ამ მონაკვეთში ცვლილების შეტანა მოგიწევთ:

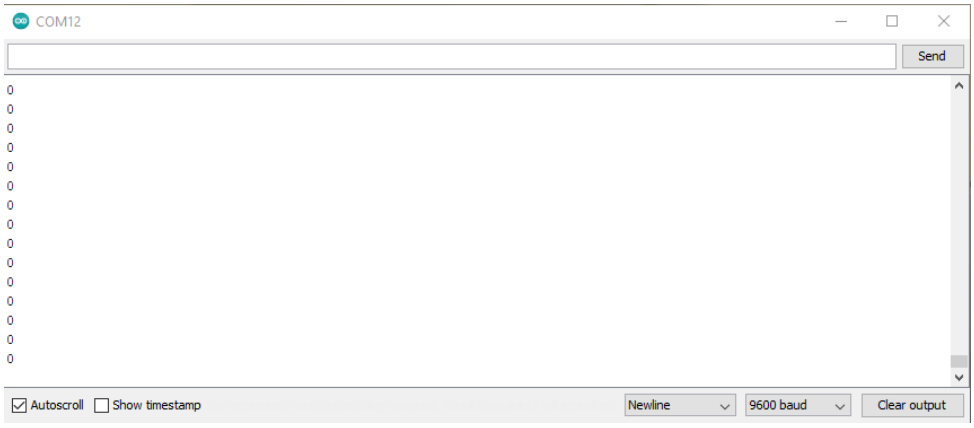
```
void loop() {
 for (int servo = 0; servo < 4; servo++ ) {
  srituhobby.setPWM(servo, 0, servoMIN);
  Serial.println(servo);
  delay(300);
 }
 for (int servo = 3; servo >= 0; servo-- ) {
  srituhobby.setPWM(servo, 0, servoMAX);
  Serial.println(servo);
  delay(300);
 }
```

თქვენ შეგიძლიათ კოდში მონიშნოთ თექვსმეტივე პორტი. ამ შემთხვევაში კოდის ზემოთ ნაჩვენები მონაკვეთი ასეთი იქნება:

```
void loop() {
 for (int servo = 0; servo < 16; servo++ ) {
  srituhobby.setPWM(servo, 0, servoMIN);
  Serial.println(servo);
  delay(300);
 }
 for (int servo = 16; servo >= 0; servo-- ) {
  srituhobby.setPWM(servo, 0, servoMAX);
  Serial.println(servo);
  delay(300);
 }
```

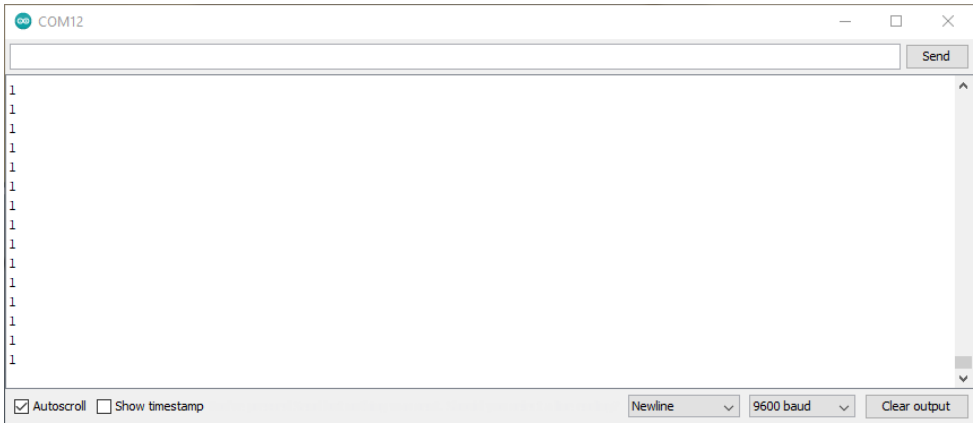
შეგიძლიათ თექვსმეტივე Servo ძრავა ერთდროულად შეამოწმოთ, ან რამდენიმე Servo ძრავით ცვალოთ პორტები.

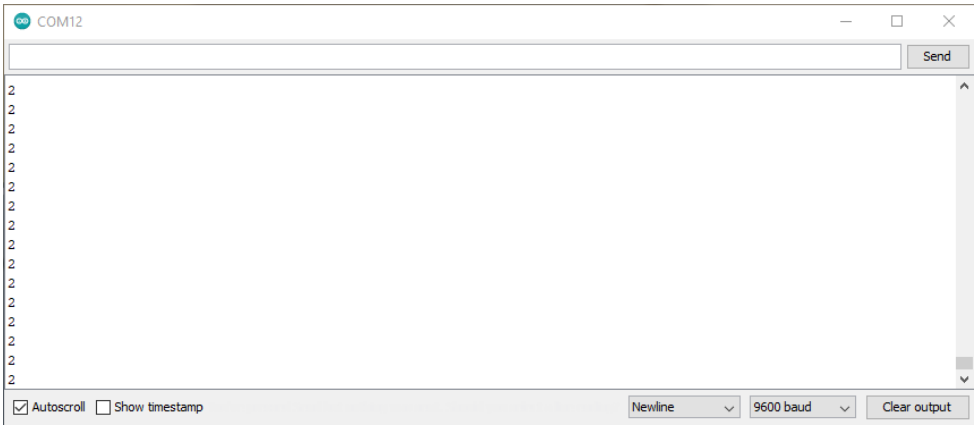
7. ასეთი მონაცემები უნდა დაფიქსირდეს თქვენს მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor) Project_10b-ს შემთხვევაში (იხ. სურათი 5):



Port 0 - Servo 1

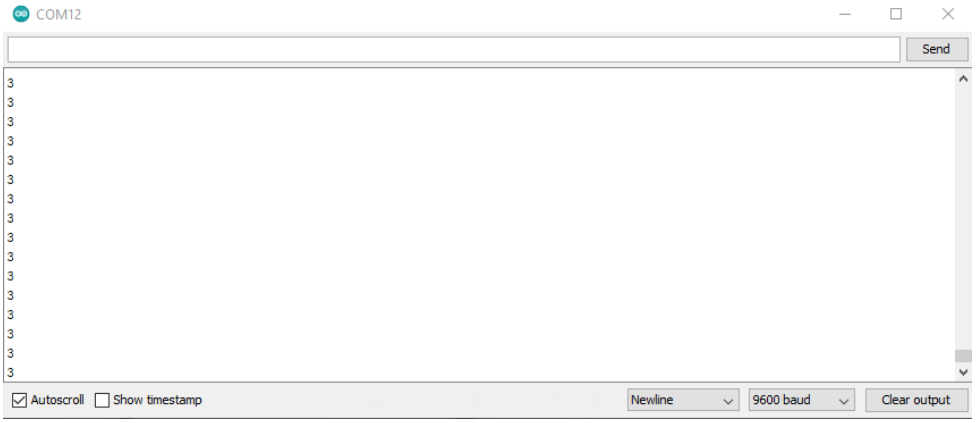
Port 1 - Servo 2





Port 2 – Servo 3

Port 3 – Servo 4



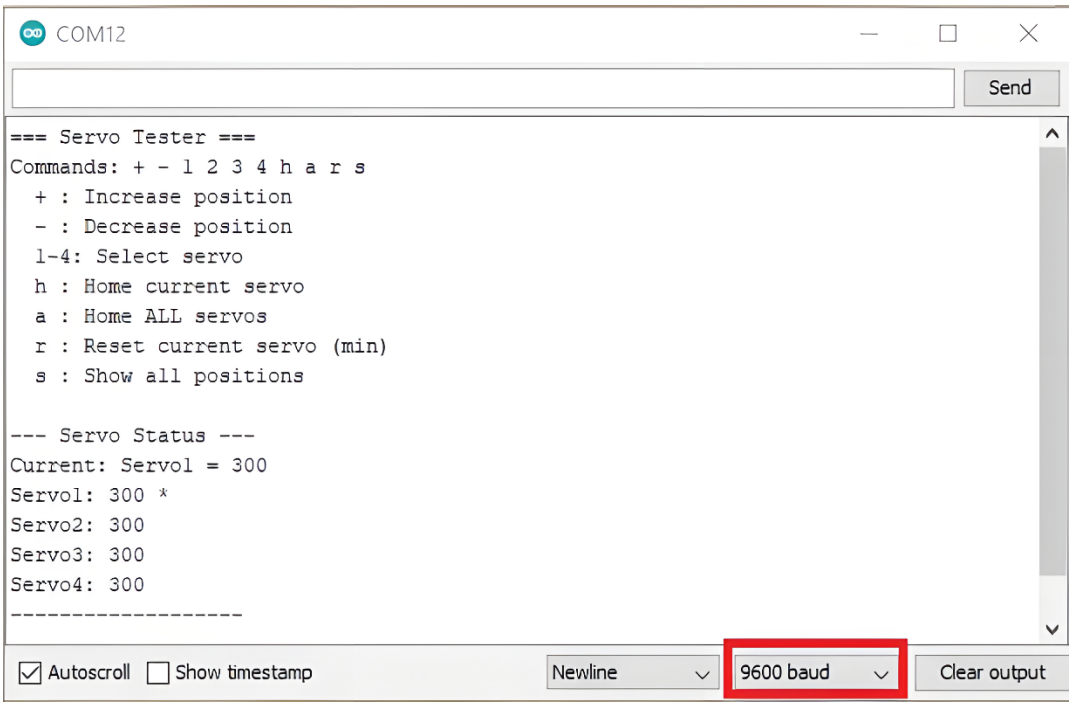
სურათი 5. Project_10b

SERVO ძრავების კალიბრაცია

ჩვენ უკვე გავეცანით PCA9685 PWM Servo ძრავას მოდულის ფუნქციონალს და მუშაობის პრინციპს, ასევე, გავტესტეთ 4 Servo ძრავა. ახლა საჭიროა ამ ძრავების კალიბრაცია - როგორ დავაყენოთ თითოეული მათგანი ჩვენთვის სასურველ პოზიციაზე. ეს ცოდნა ჩვენ გამოგვადგება მომდევნო ორი პროექტის შექმნის დროს. სქემა (იხ. სურათი 2) და კომპონენტები იგივე რჩება.

1. დააკავშირეთ არდუინო კომპიუტერთან და ჩატვირთეთ კოდი Project_10c.

2. გახსენით მიმდევრობითი კომუნიკაციის მონიტორი (Serial Monitor), baud დააყენეთ 9600-ზე. მიიღებთ შემდეგ ინფორმაციას (იხ. სურათი 6):



სურათი 6. Servo ტესტერი

რას გვეუბნება ეს ინფორმაცია:

=== Servo ტესტერი ===

Commands: **+ - 1 2 3 4 h a r s** - ბრძანებები

+ : Increase position - პოზიციის გაზრდა

- : Decrease position - - პოზიციის შემცირება

1-4: Select servo - Servo ძრავას არჩევა

h : Home current servo - მიმდინარე Servo ძრავას პოზიცია (მაგალითად, h3-ის გაშვება მესამე Servo ძრავას პოზიციას გამოიტანს ეკრანზე)

a : Home ALL servos - a - ყველა Servo ძრავას საწყის პოზიციაზე დაბრუნება

r : Reset current servo (min) - მიმდინარე Servo ძრავას მინიმალურ პოზიციაზე დაყენება

s : Show all positions - ყველა Servo ძრავას პოზიციის ჩვენება

--- Servo Status ---

Current: **Servo1** = 300 მიმდინარე Servo ძრავა

Servo1: 300 *

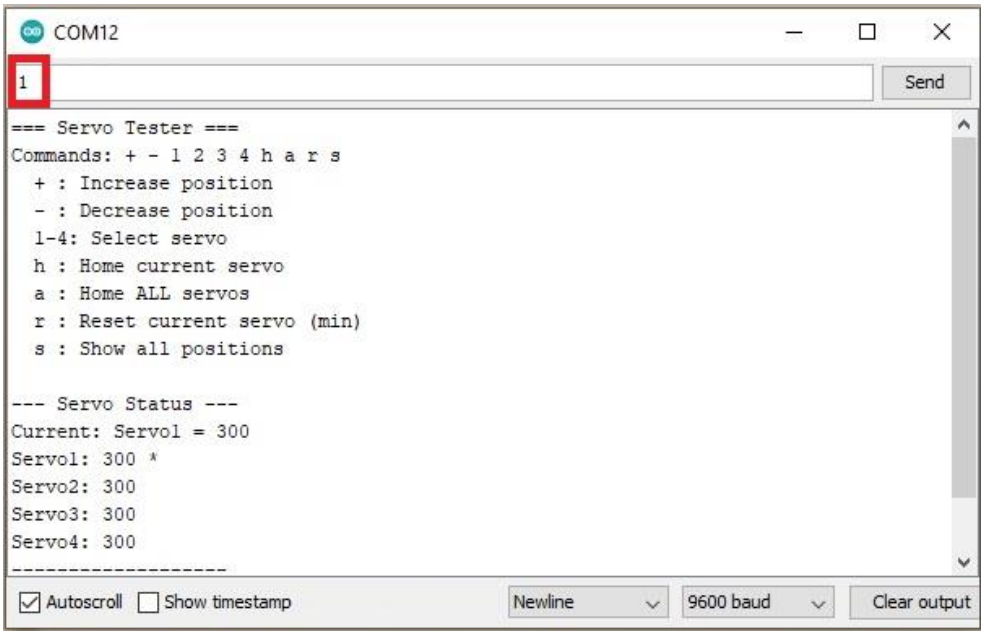
Servo2: 300

Servo3: 300 ოთხივე Servo ძრავას საწყისი პოზიცია

Servo4: 300

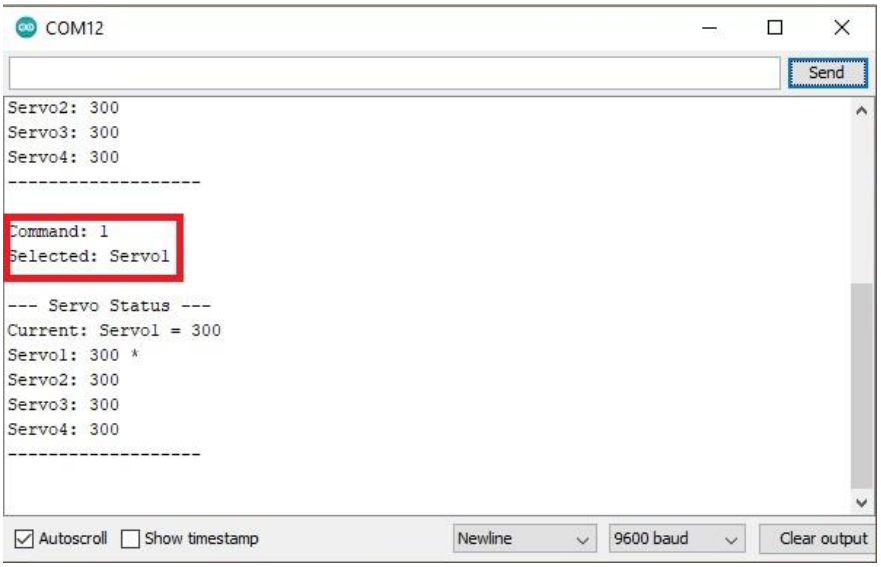
რომელ ბრძანებასაც გაუშვებთ მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor), იმის შესაბამის ინფორმაციას მიიღებთ. ოთხივე Servo ძრავას საწყისი პოზიცია უკვე ვიცით. ახლა, ვნახოთ, მაგალითად, როგორ გავზარდოთ Servo1-ის პოზიცია - ცვლილებები ხდება 10-ობით (იხ. სურათი 7 - ა, ბ, გ, დ):

1. მიმდევრობითი კომუნიკაციის მონიტორის საძიებო ველში ავკრიფოთ 1 (Servo1) და გასაგზავნად დავაჭიროთ Send ღილაკს:



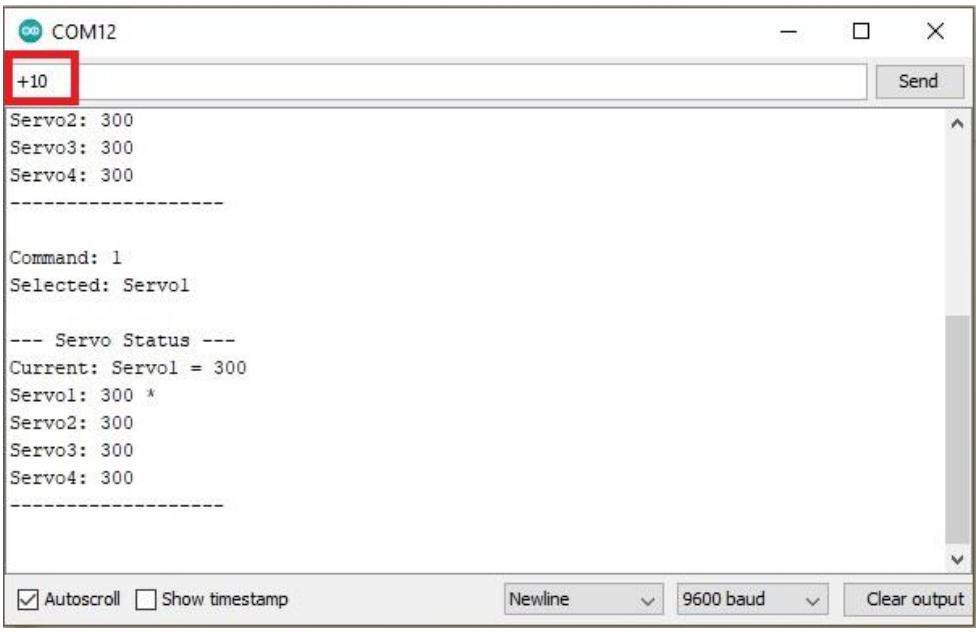
სურათი 7ა. ბრძანება პირველი Servo ძრავას პოზიციის ცვლილების შესახებ

მომდევნო სურათზე (7ბ) ვხედავთ, რომ ჩვენი ბრძანება აისახა მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor):



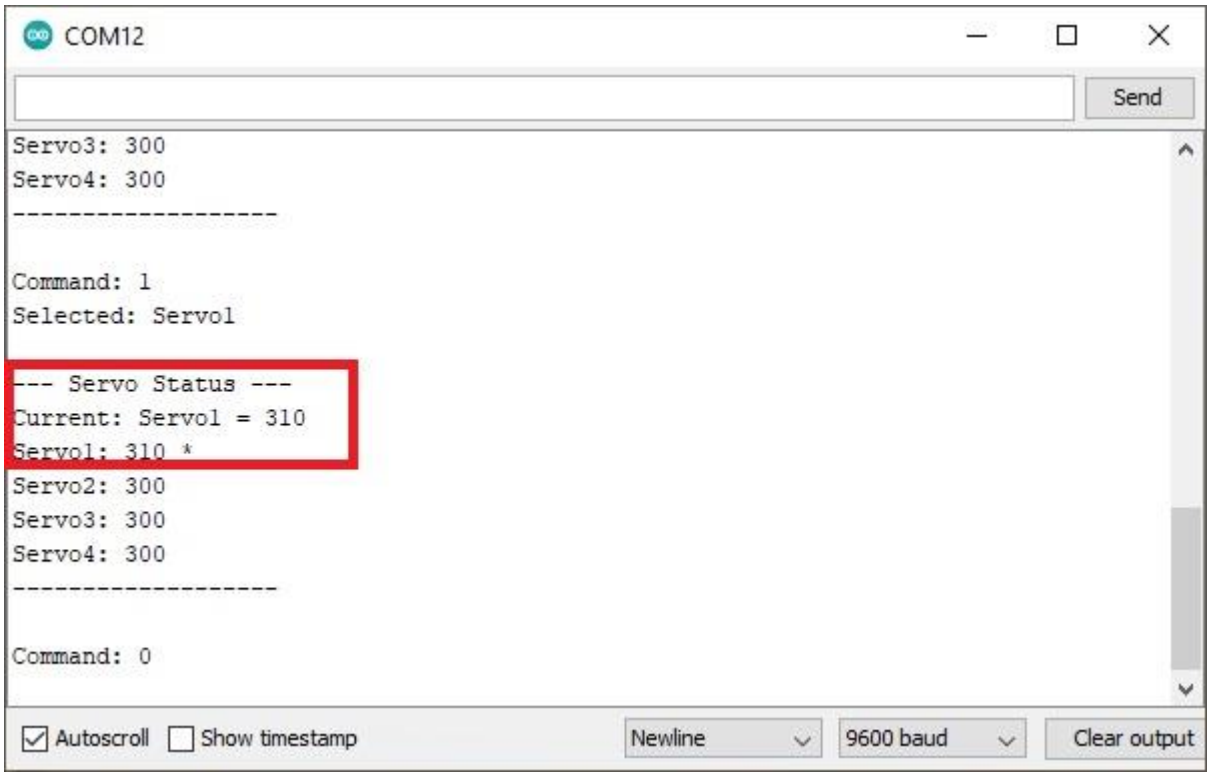
სურათი 7ბ. ბრძანება ასახულია მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor)

2. ახლა ავკრიფოთ +10 და დავაჭიროთ ისევ Send ღილაკს. ვნახოთ, რა მოხდება:



სურათი 7გ. პოზიციის +10-ით გაზრდის ბრძანება

მომდევნო სურათზე (7დ) ვხედავთ, რომ ჩვენი ბრძანება აისახა მიმდევრობითი კომუნიკაციის მონიტორზე (Serial Monitor) და Servo1-ის პოზიციაც შეიცვალა - იყო 300 და გახდა 310:



სურათი 7გ. Servo1-ის პოზიცია შეიცვალა

მსგავსად ვმოქმედებთ პოზიციის შემცირების შემთხვევაში - იმ განსხვავებით, რომ ბრძანება იქნება -10.

მეორე და დანარჩენი Servo ძრავების პოზიციის ცვლილებისას ჯერ ვკრიფავთ Servo ძრავას რიგით ნომერს, მაგალითად, 3 და ვაგზავნით ბრძანებას, რომ მოინიშნოს Servo3, შემდეგ მისი პოზიციის შესაცვლელად: -10 ან +10 (ზუსტად ისე, როგორც პირველი Servo ძრავას დროს).

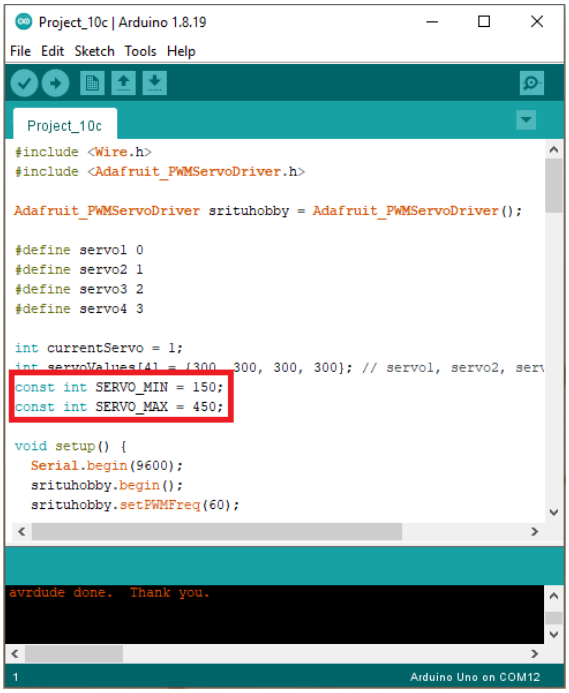
გასათვალისწინებელია!

Servo1-ის შემთხვევაში ჩვენ მხოლოდ ერთხელ ვაფიქსირებთ მისი მონიშვნის ბრძანებას - 1 და ვაგრძელებთ მისი პოზიციის ცვლილებას იქამდე, სადამდეც გვჭირდება (მაგ. -10, -10, 10). ხოლო დანარჩენი Servo ძრავების შემთხვევაში, პოზიციის ყოველი ცვლილების წინ ვაგზავნით ჯერ Servo ძრავის ნომერს, შემდეგ პოზიციის ცვლილების ბრძანებას. მაგალითად, გვინდა Servo2-ის პოზიციის ცვლილება:

ჯერ ვაგზავნით 2-ს, შემდეგ -10, ისევ, ჯერ 2-ს, შემდეგ -10 და ასე მანამდე, ვიდრე სასურველ პოზიციას არ დავაფიქსირებთ.

მნიშვნელოვანია!

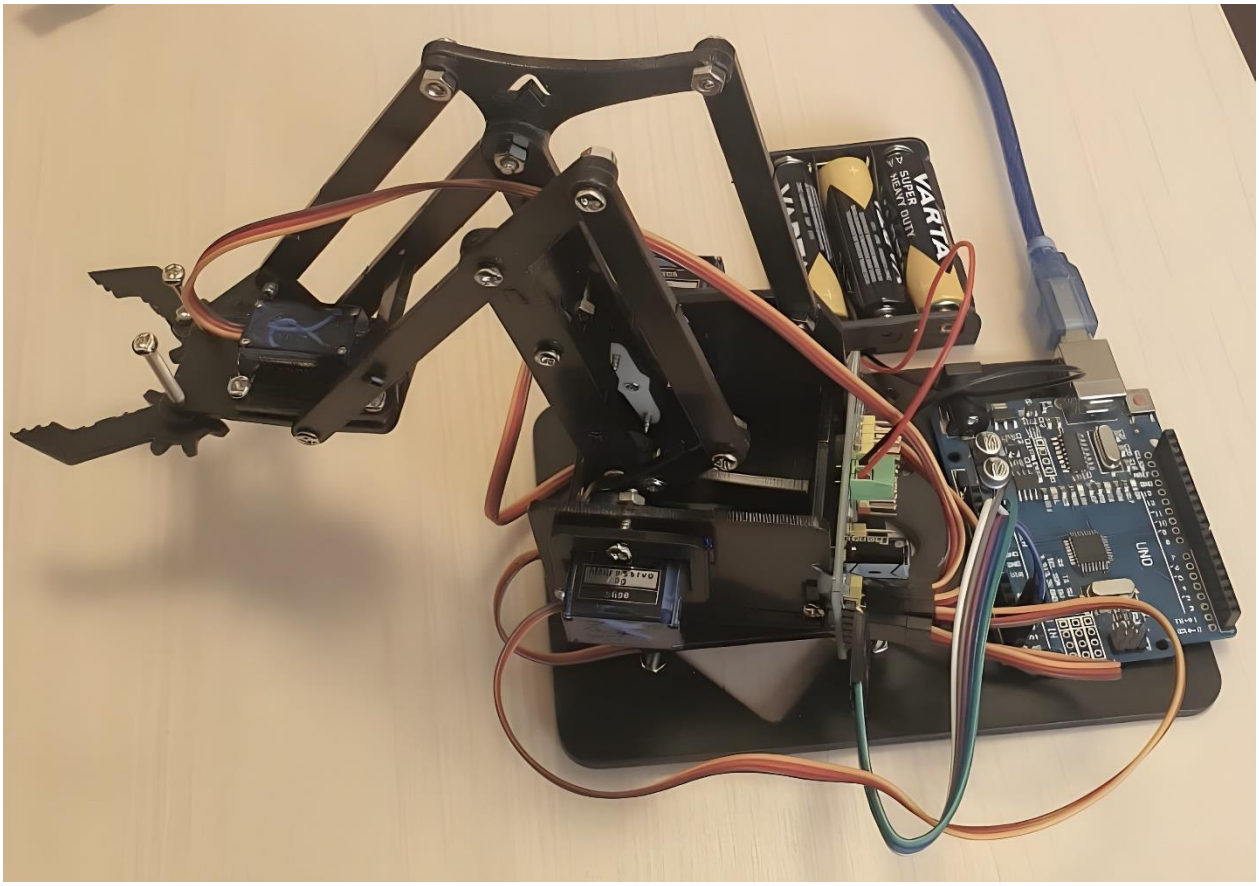
როგორც ხედავთ, ჩვენს კოდში მითითებული Servo ძრავების მოძრაობის დიაპაზონი ლიმიტირებულია (იხ. სურათი 8).



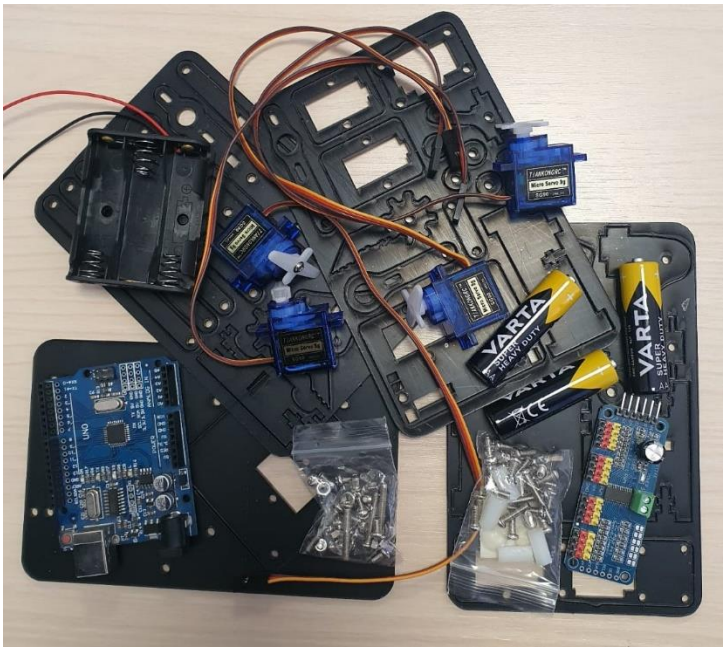
სურათი 8. Servo ძრავების მოძრაობა ლიმიტირებულია

ამიტომ, გაითვალისწინეთ პოზიციების ცვლილებისას.

რობოტი მკლავი



საჭირო რესურსი



- Arduino UNO
- PCA9685 PWM Servodრავას დრაივერი

  (16 არხიანი 12-ბიტიანი)
- Servo ძრავა Tower Pro SG90 9g x 4
- 3 ცალი 1.5V ბატარეა და ბუდე
- რობოტი მკლავი
- დედალ-მამალი გამტარები

საჭირო ბიბლიოთეკა

- PWMServoDriver
- Wire

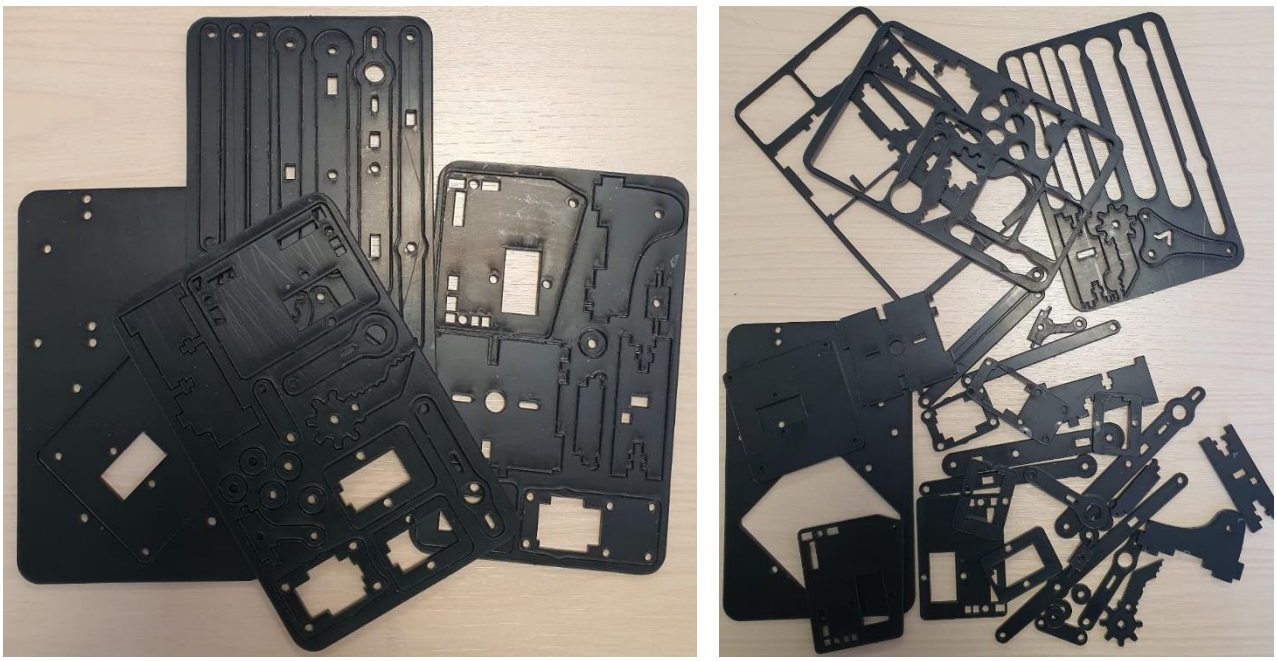
რობოტი მკლავი

რობოტი მკლავი დაპროგრამებული მექანიკური სისტემაა. ის ფუნქციონირებს როგორც ჩვეულებრივი ადამიანის ხელი. რობოტი მკლავები სხვადასხვა ზომისაა, პატარადან ძალიან დიდამდე. ის გამოიყენება ინდუსტრიაში და შეუძლია შეასრულოს ისეთი დავალებები, რაც ნორმალური ადამიანისთვის შეუძლებელია. ბაზარზე ხელმისაწვდომია რობოტი მკლავების ფართო სპექტრი, რომლებიც შექმნილია სახლში ყოველდღიური ამოცანების გასაადვილებლად. ეს სახელმძღვანელო ეტაპობრივად აღწერს, თუ როგორ უნდა ააწყოთ მექანიზმი და როგორ მუშაობს ის Arduino-სთან. ეს სახელმძღვანელო ასევე დაგეხმარებათ გაიგოთ, თუ როგორ მუშაობს რობოტი მკლავი.

მოდით, ეტაპობრივად, ნაბიჯ-ნაბიჯ ავაწყოთ ეს პროექტი. ამ პროექტით ჩვენ ვისწავლით რობოტი მკლავის აწყობასა და მართვას Tower Pro SG90 9g 4 ცალი Servo ძრავას საშუალებით. X პროექტში უკვე გავარკვიეთ Servo ძრავების ტესტირება და კალიბრაცია (გვ. 103).

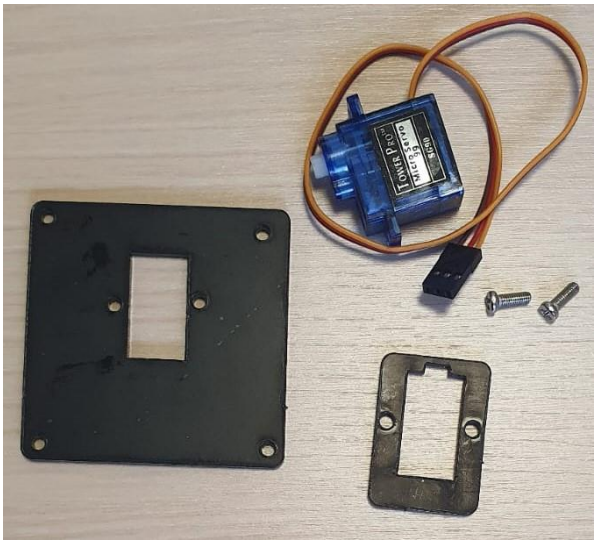
ახლა დავიწყოთ მკლავის აწყობა:

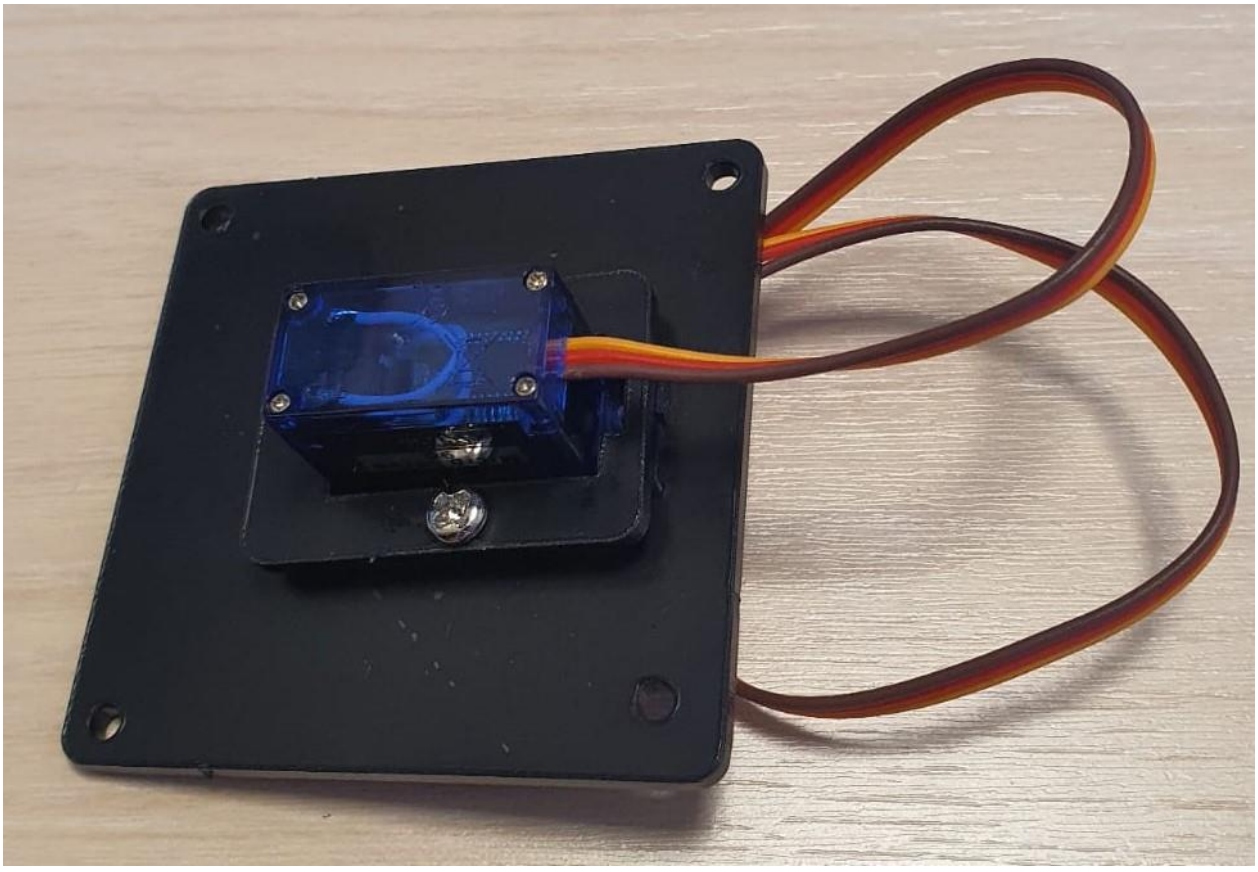
1. დაშალეთ რობოტი მკლავის კომპლექტი შემადგენელ ნაწილებად (იხ. სურათი 1):



სურათი 1.

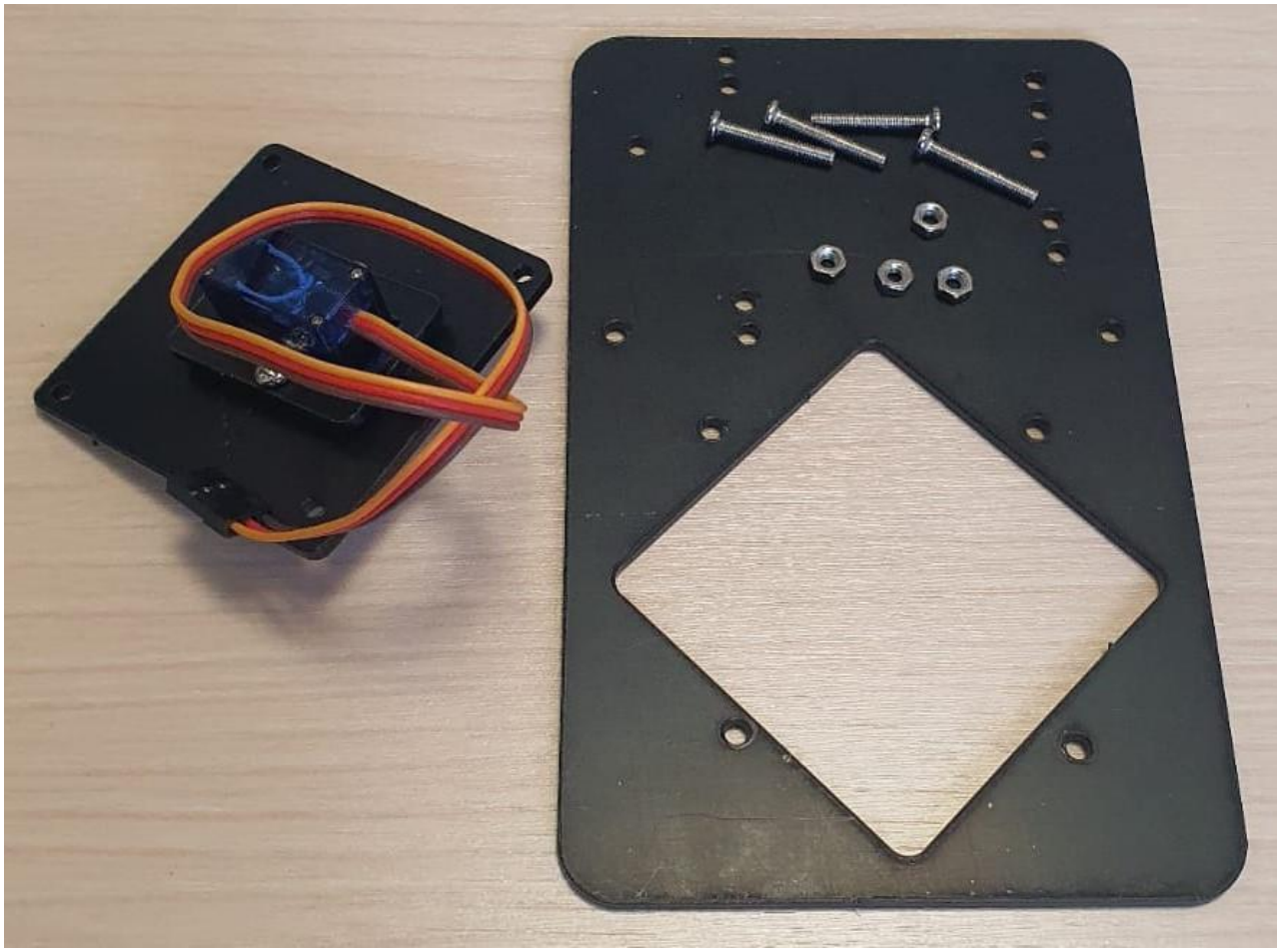
2. დააკავშირეთ Servo1 შემდეგნაირად (იხ. სურათი 2):

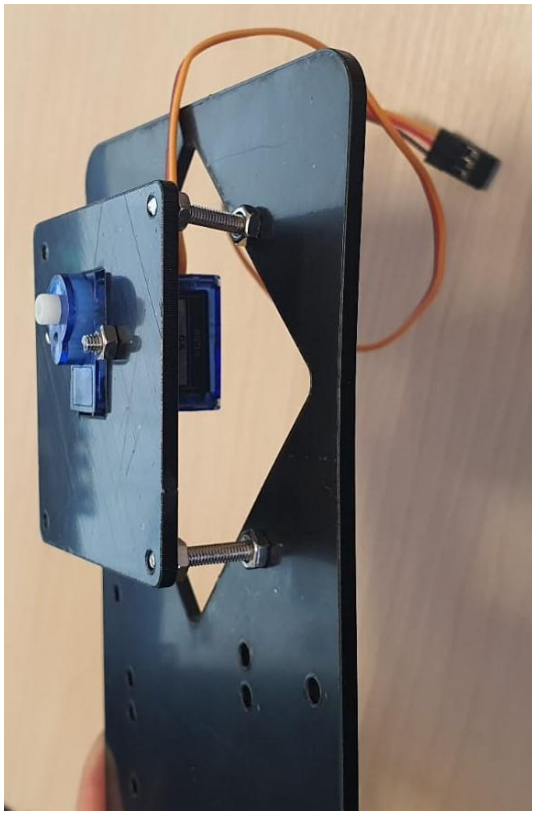


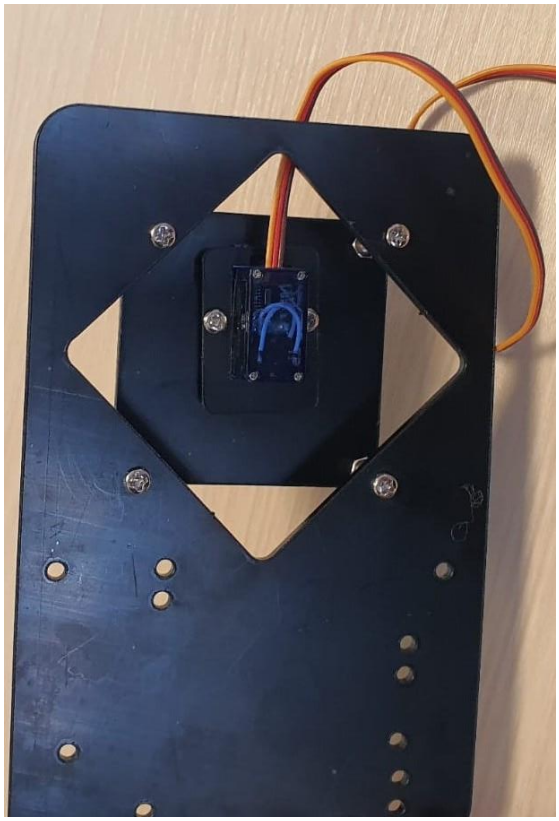


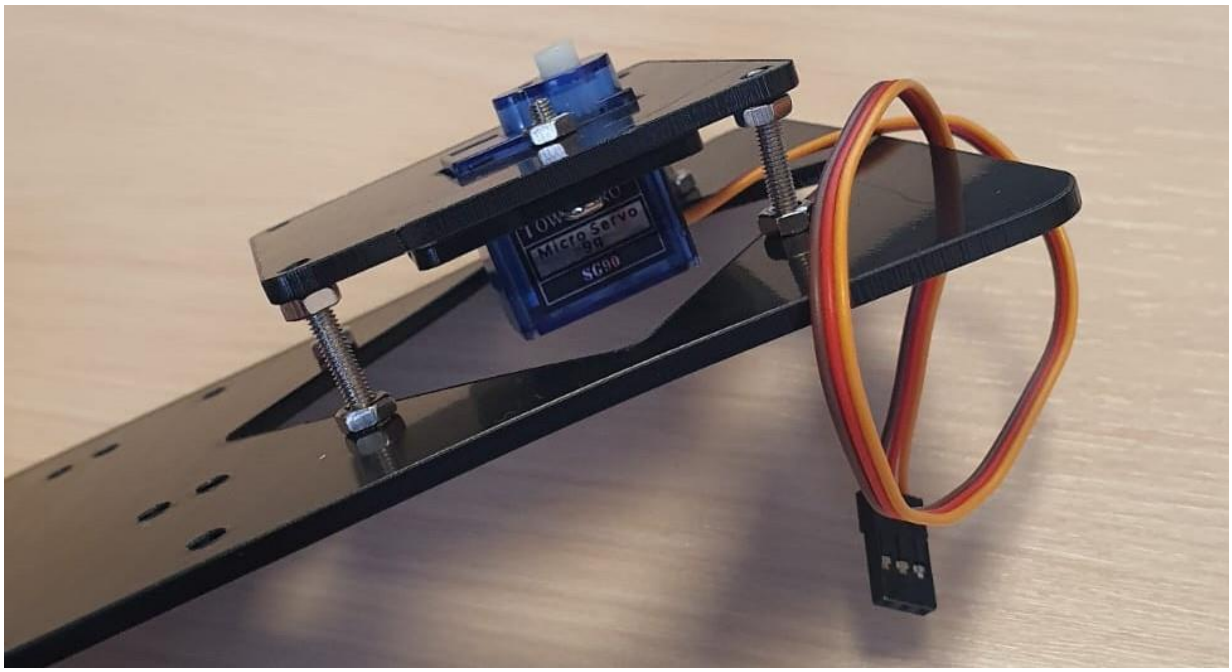
სურათი 2.

3. დააკავშირეთ საყრდენი ნაწილი Servo1-ს (იხ. სურათი 3):



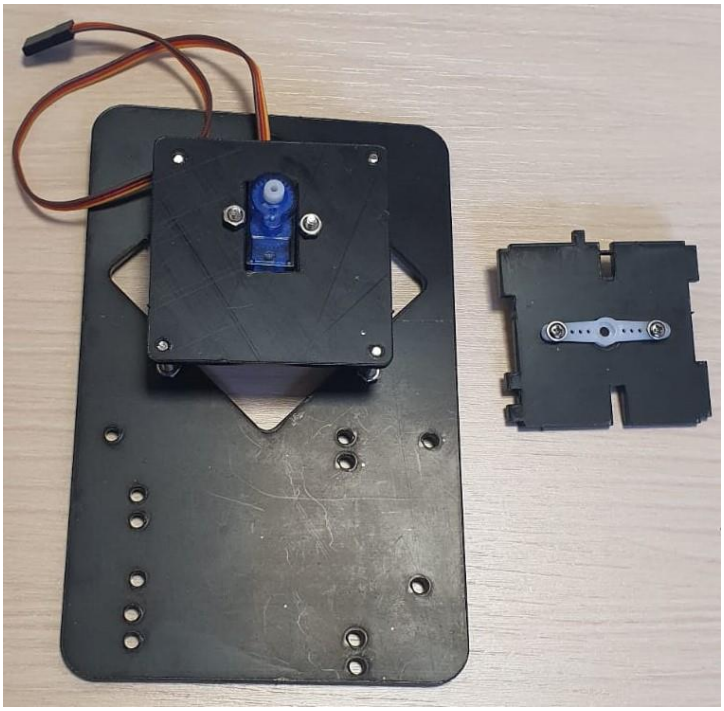


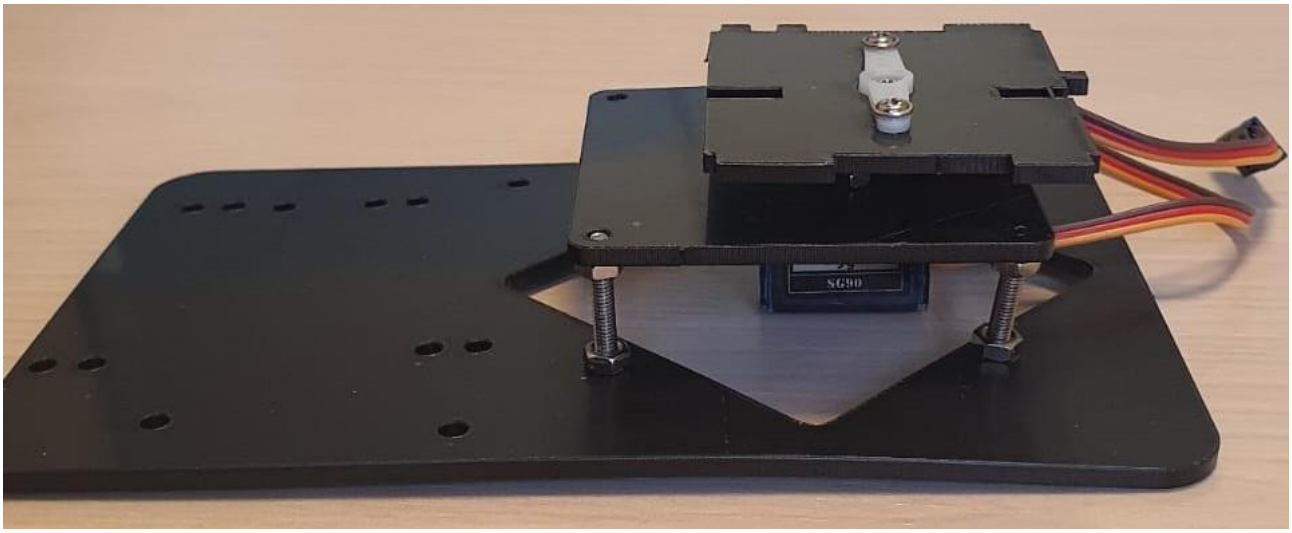




სურათი 3.

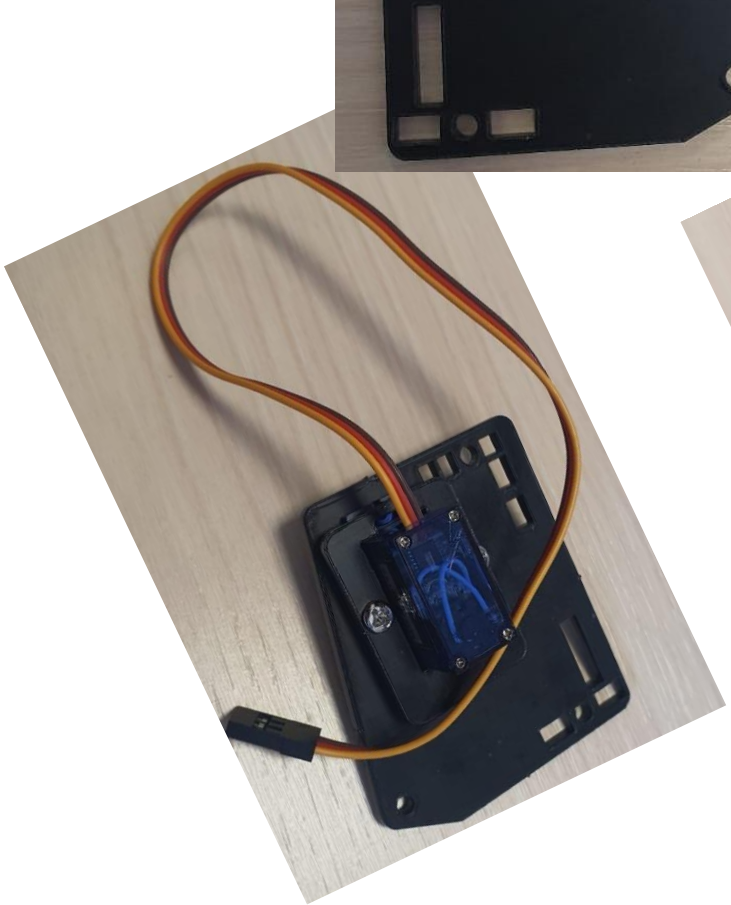
4. დაამაგრეთ Servo1-ის ფრთა (იხ. სურათი 4):

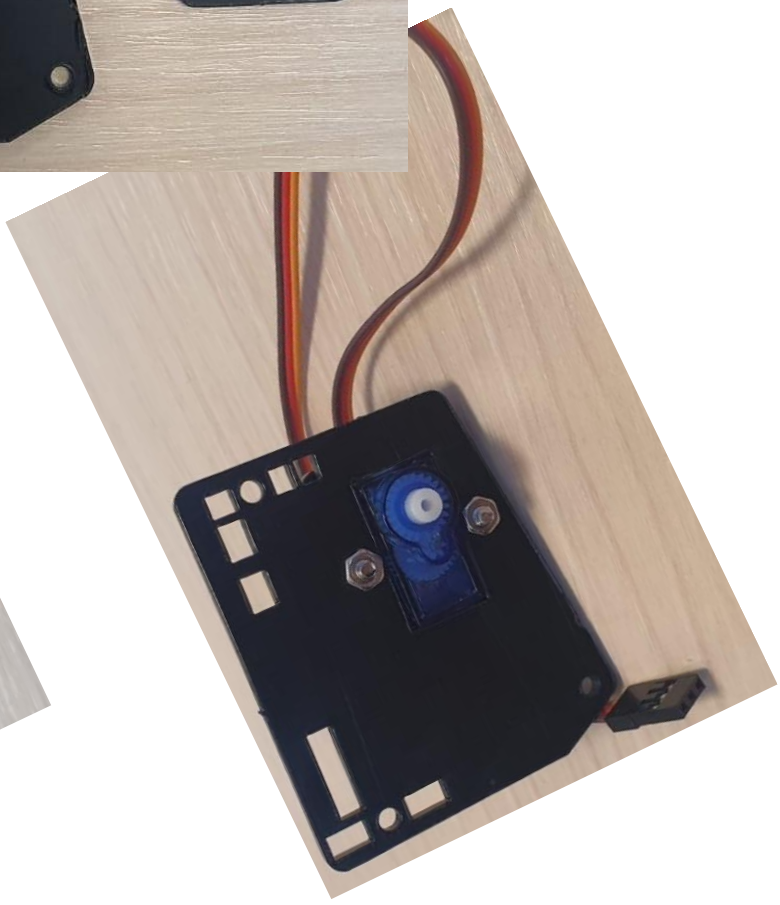


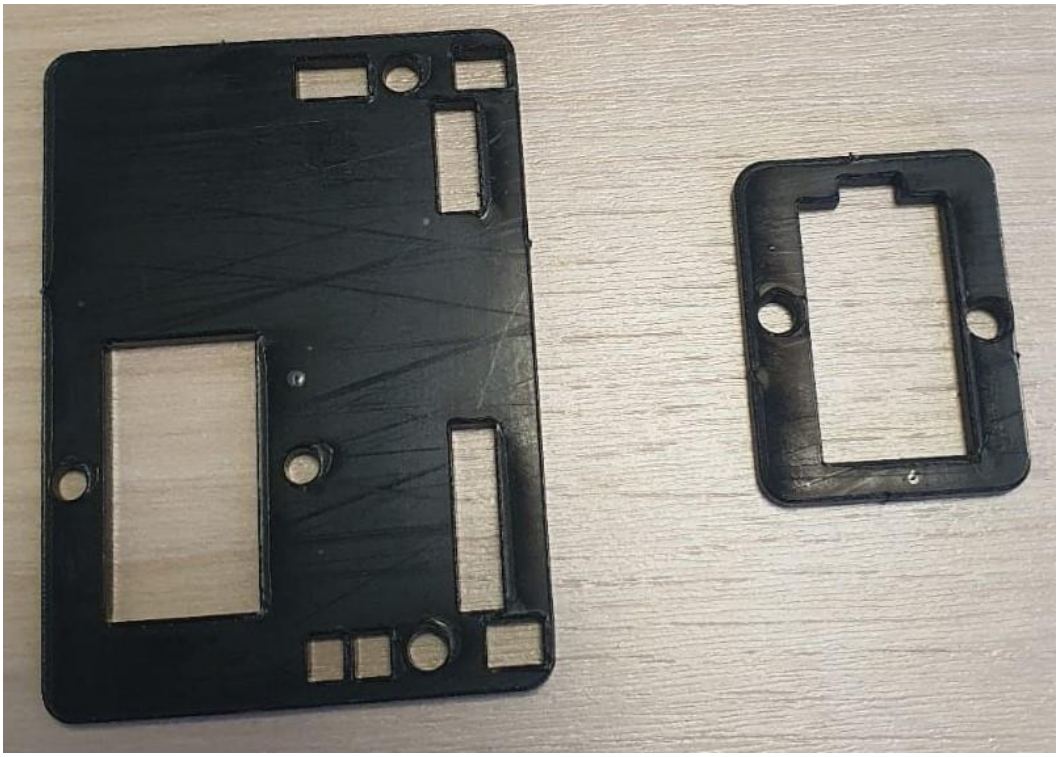


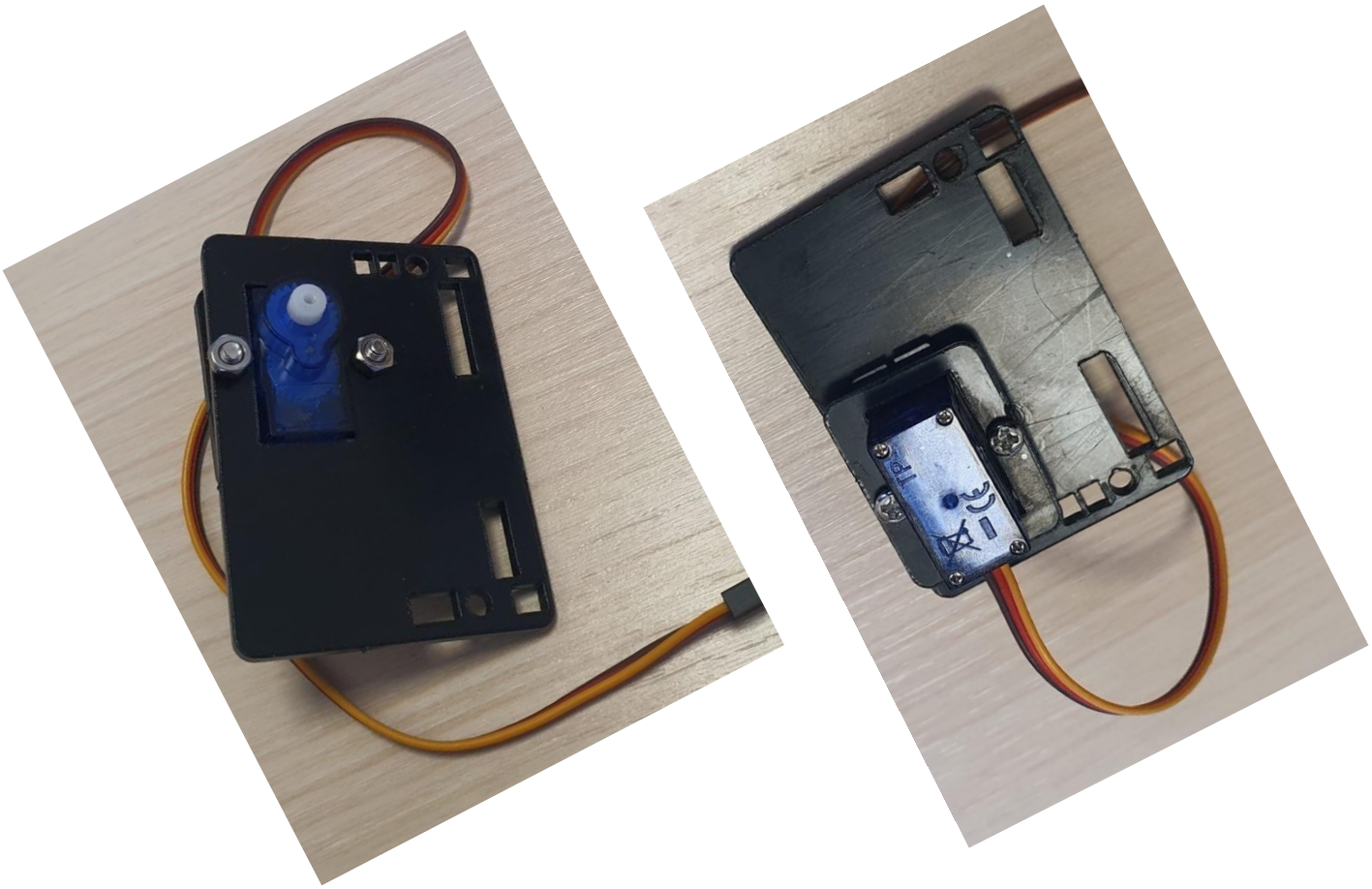
სურათი 4.

5. მოვამზადოთ გვერდითი Servo ძრავები - Servo2 და Servo3 (იხ. სურათი 5):



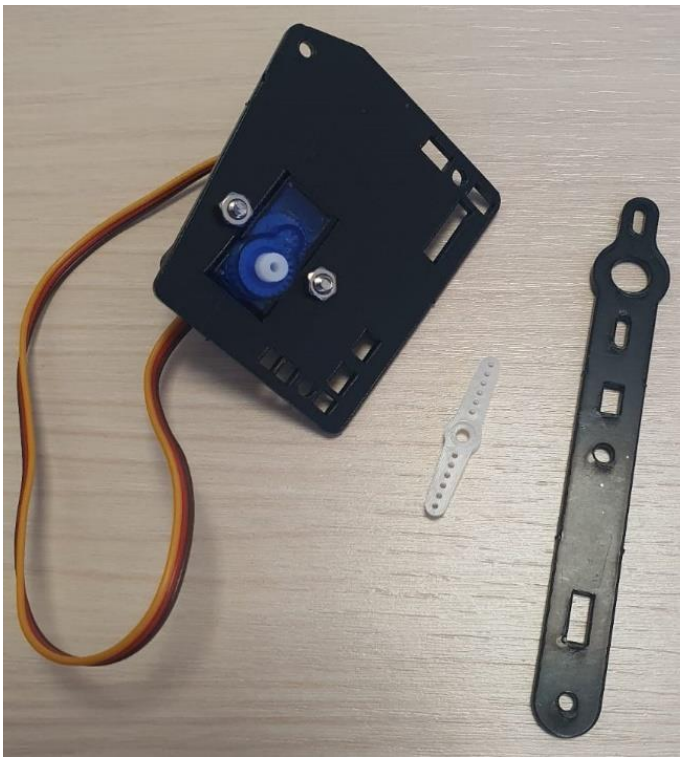


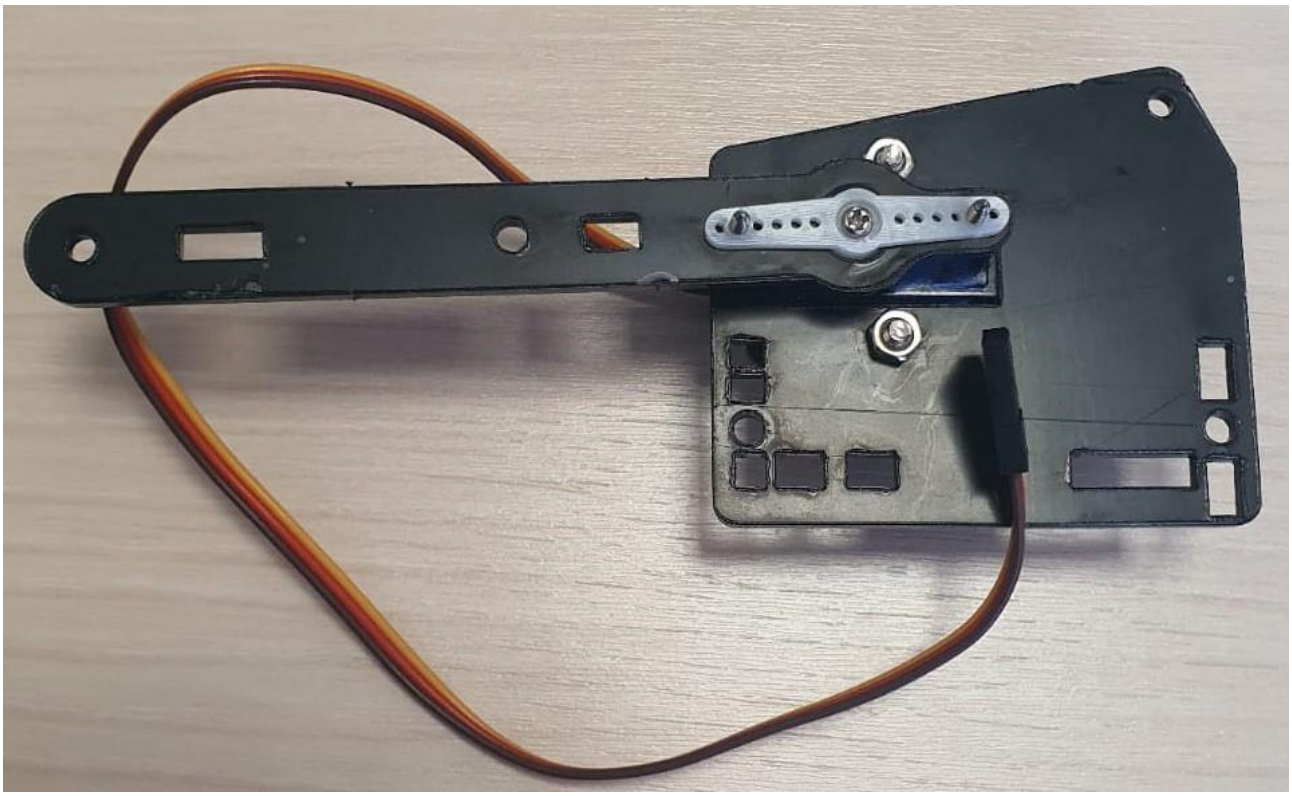


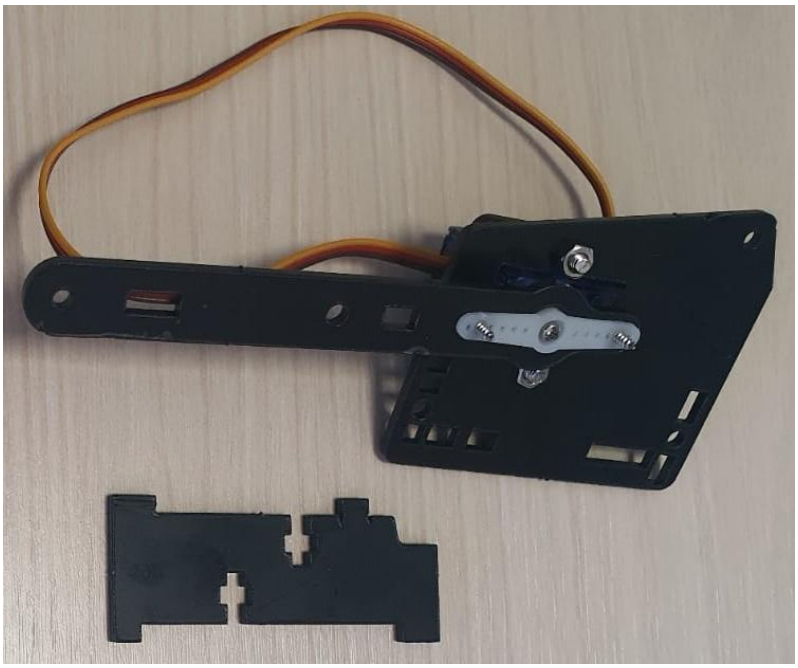


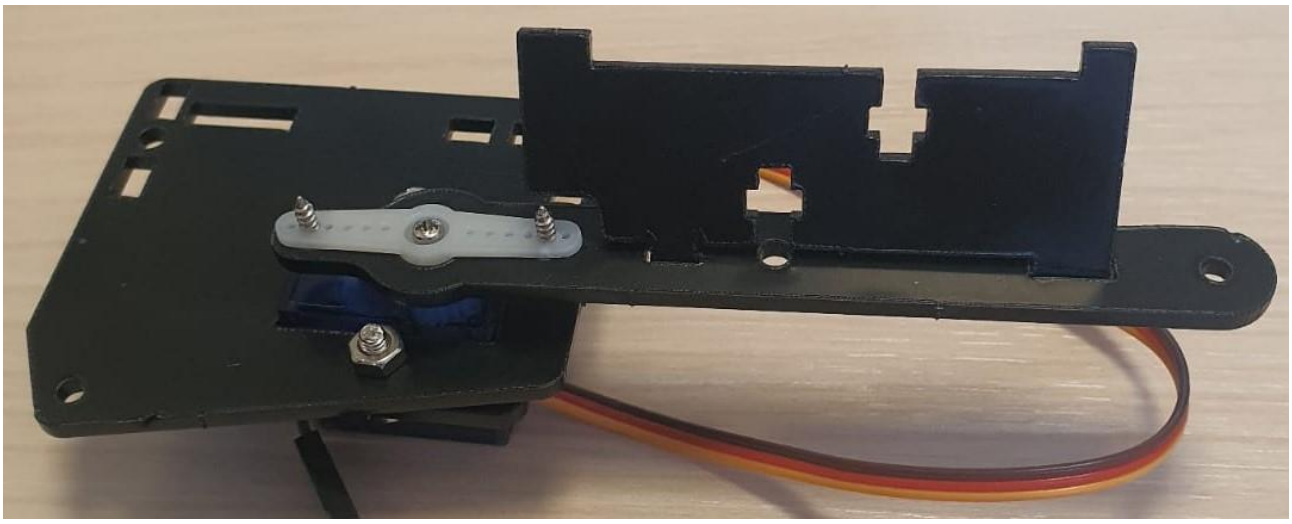
სურათი 5.

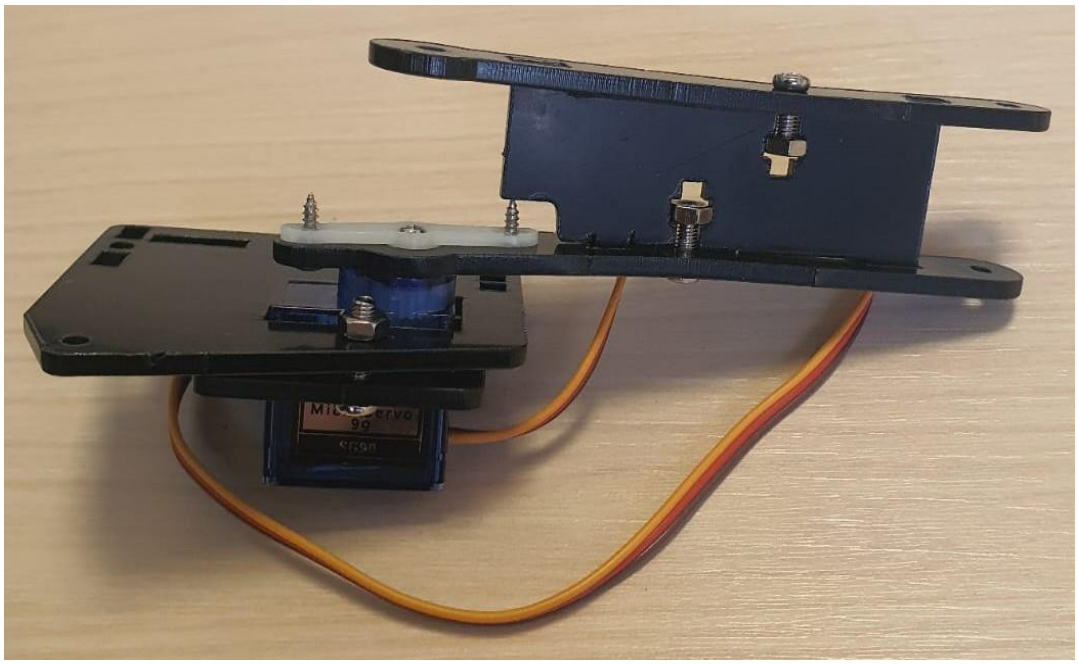
6. დაამაგრეთ მარცხენა Servo ძრავას ბრუნვის დეტალები (იხ. სურათი 6):

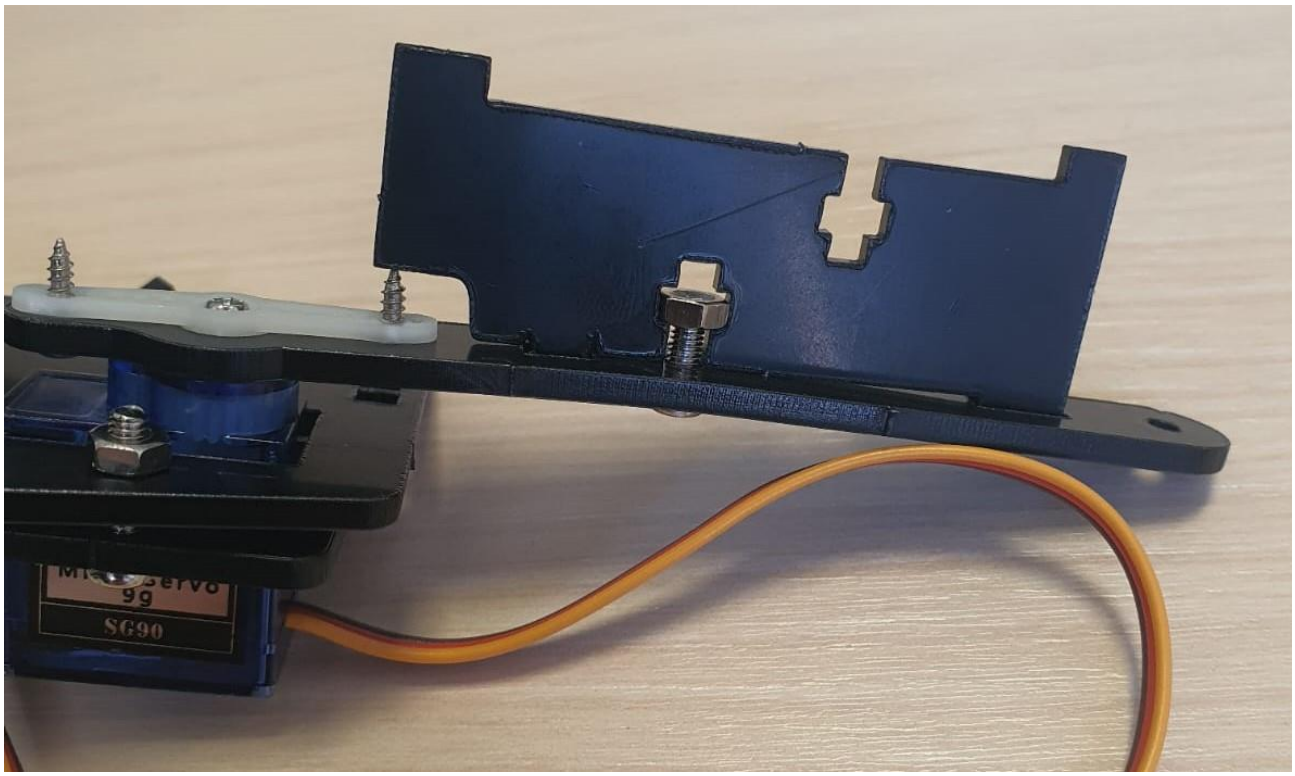


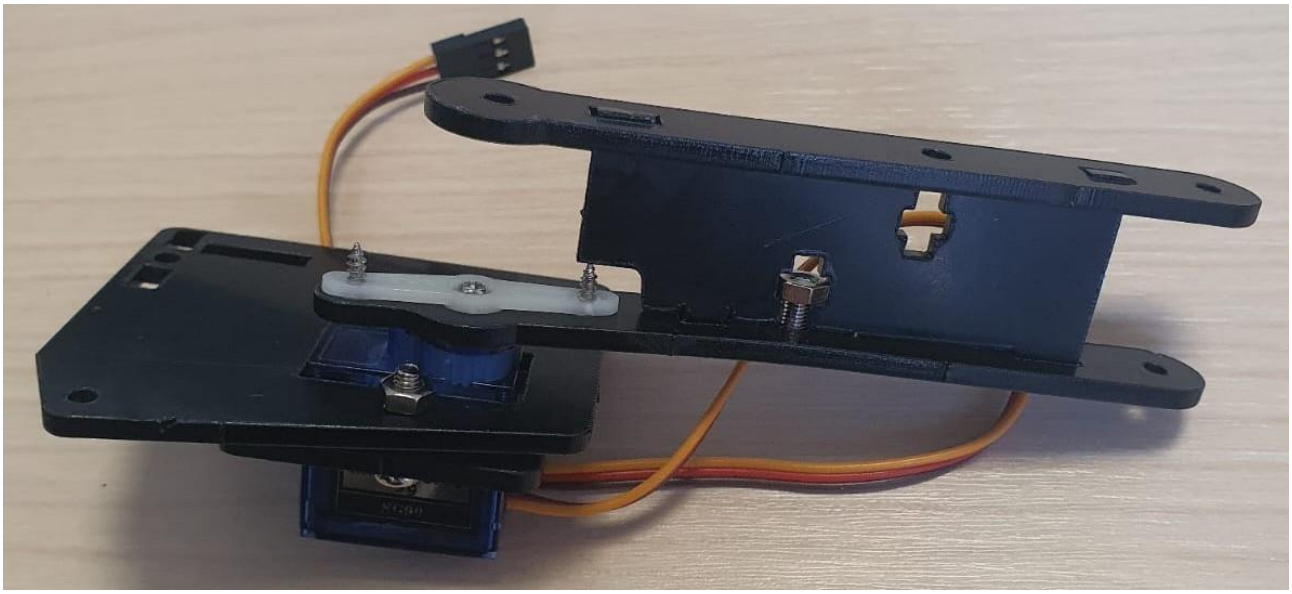










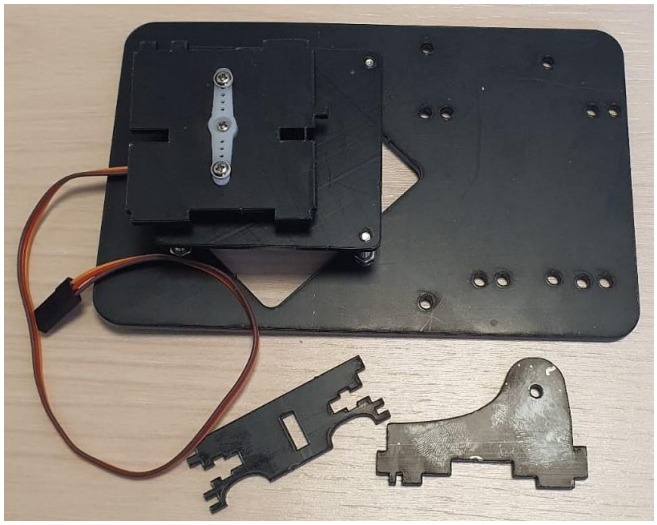


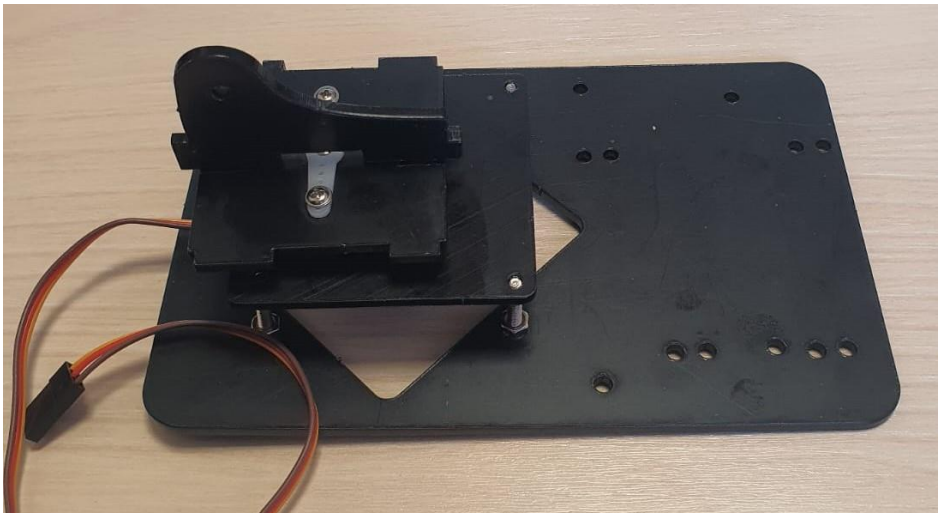
სურათი 6

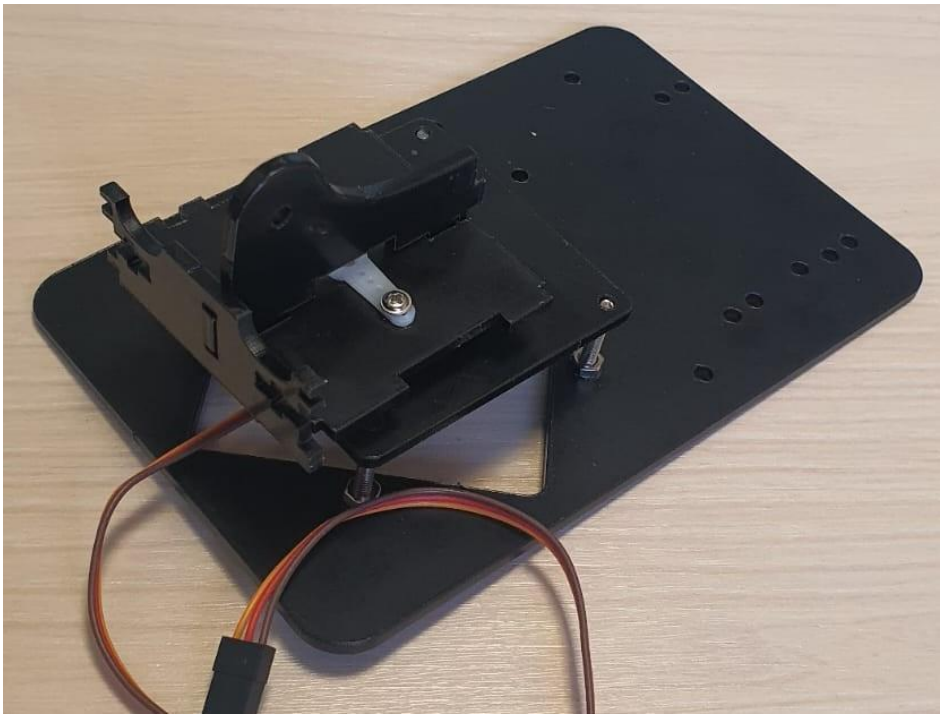
7. დაამაგრეთ მარჯვენა Servo ძრავას ბრუნვის დეტალები (იხ. სურათი 7):

სურათი 7.

8. ახლა, ქვემოთ მოცემული დეტალები დააკავშირეთ ძირითად მბრუნავ ნაწილს (იხ. სურათი 8):

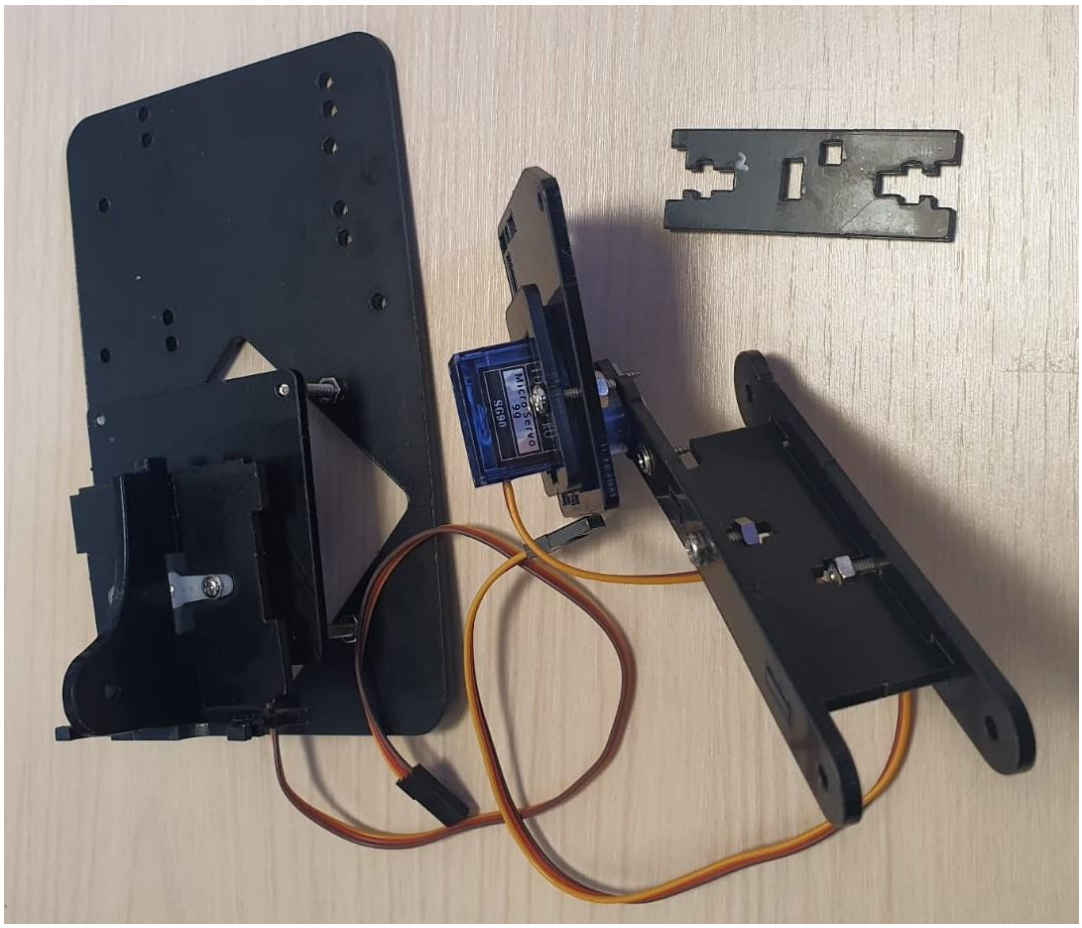


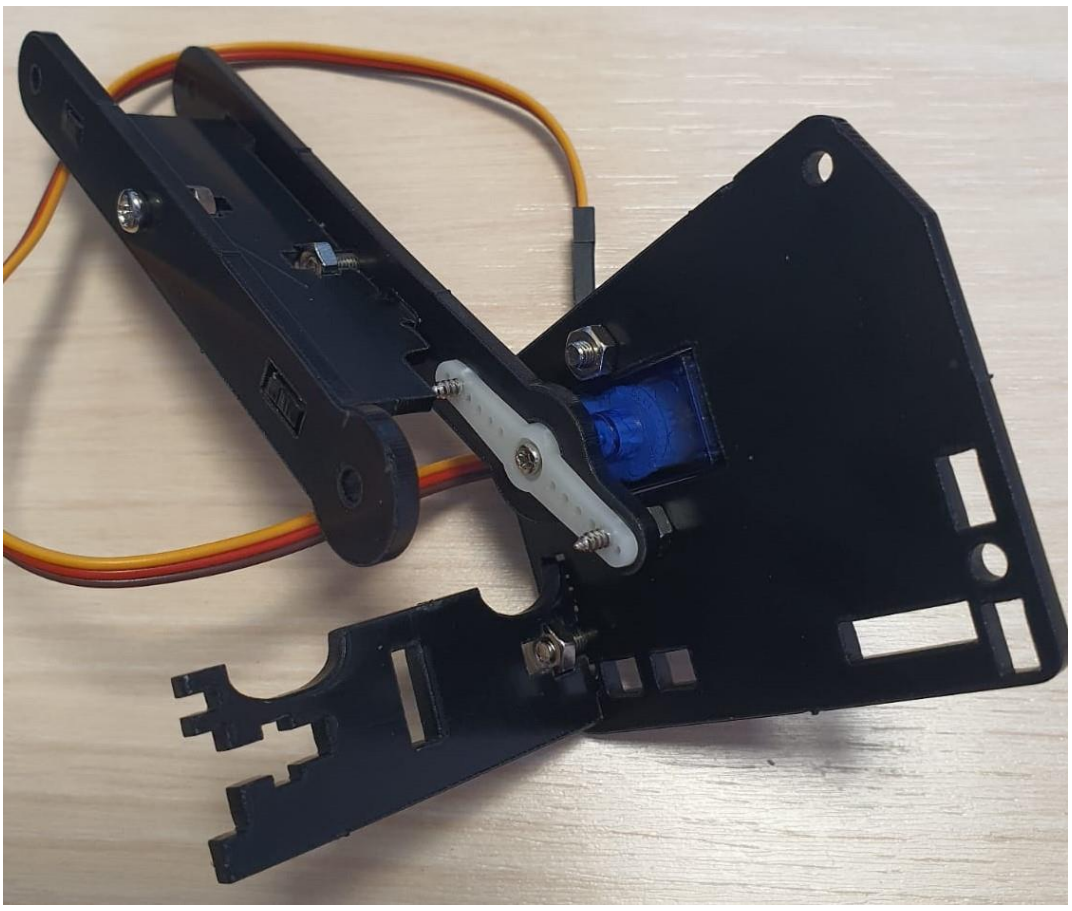


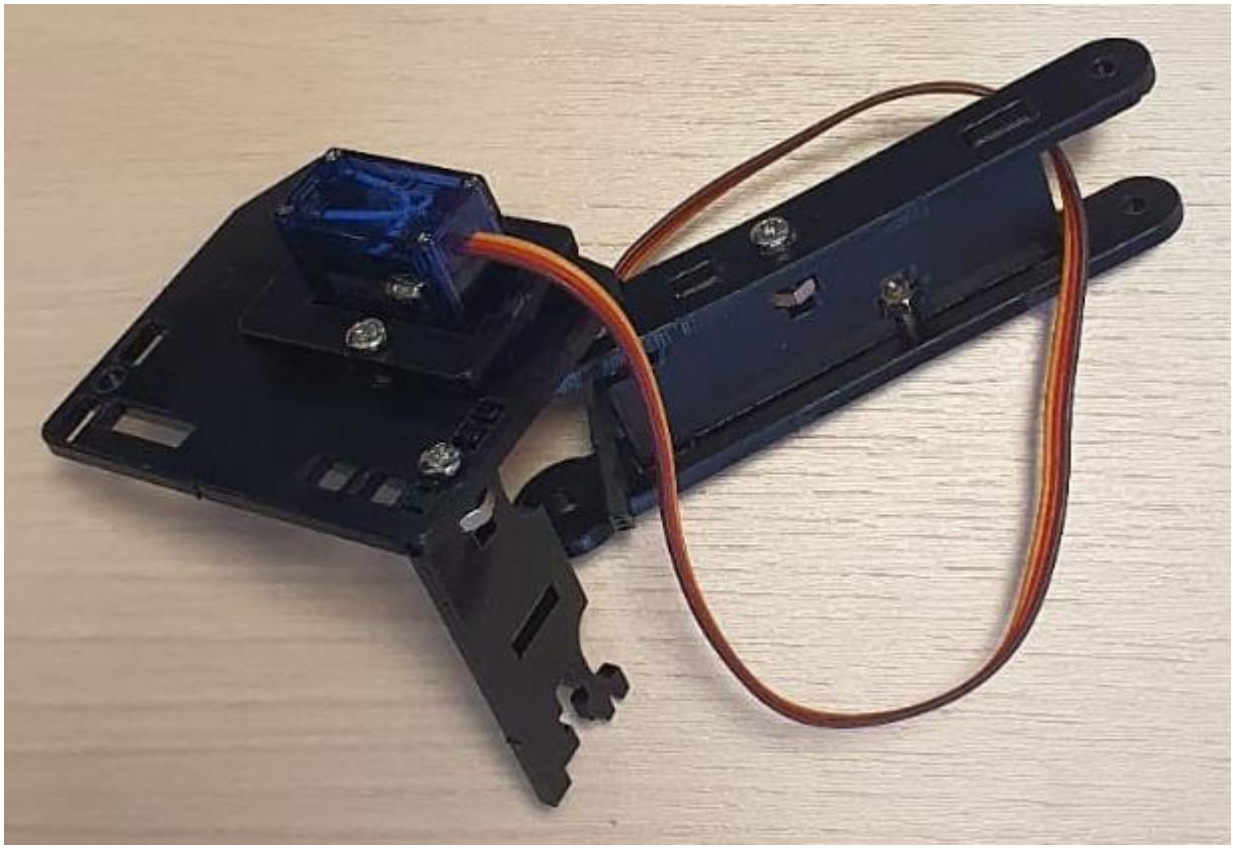


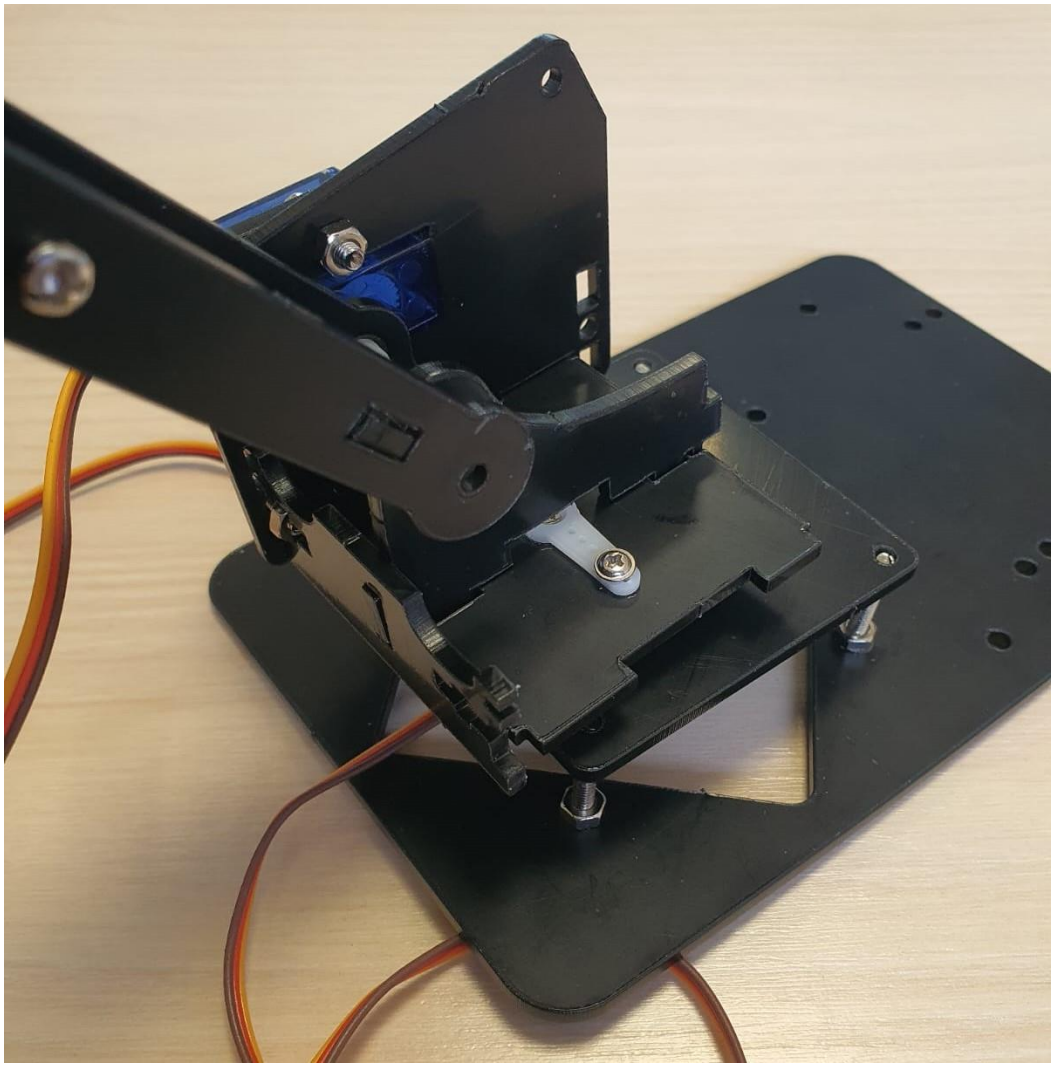
სურათი 8.

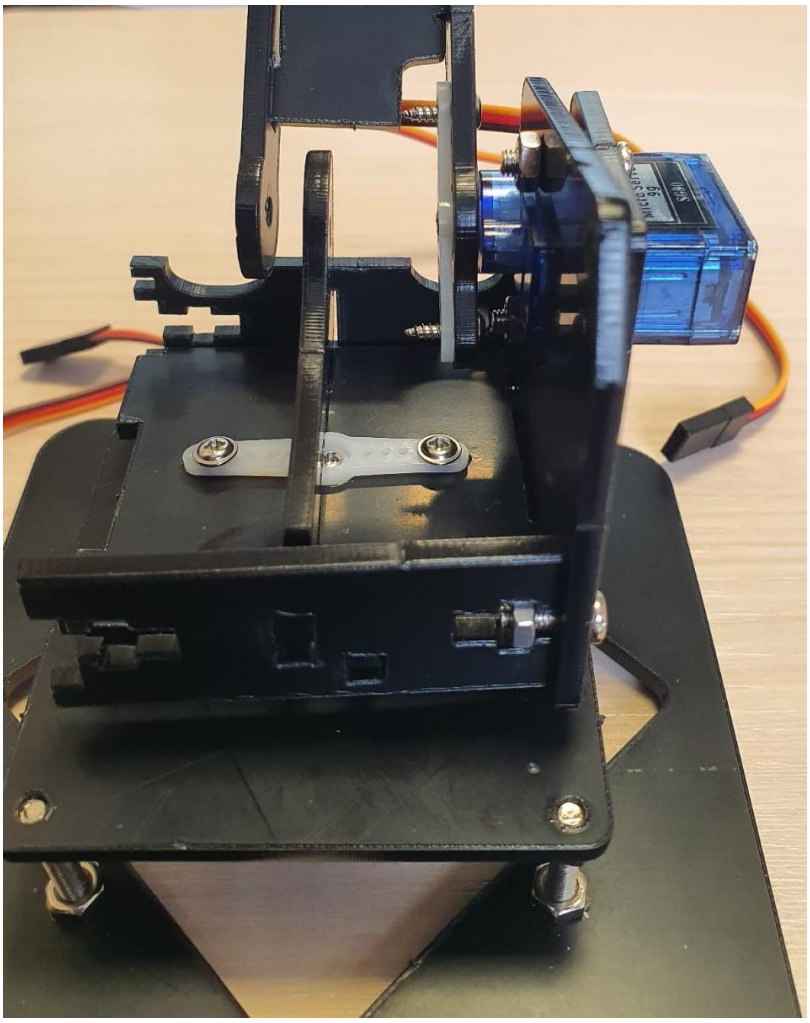
9. დაამაგრეთ მარცხენა Servo ძრავა ქვედა/საყრდენ კორპუსზე (იხ. სურათი 9):

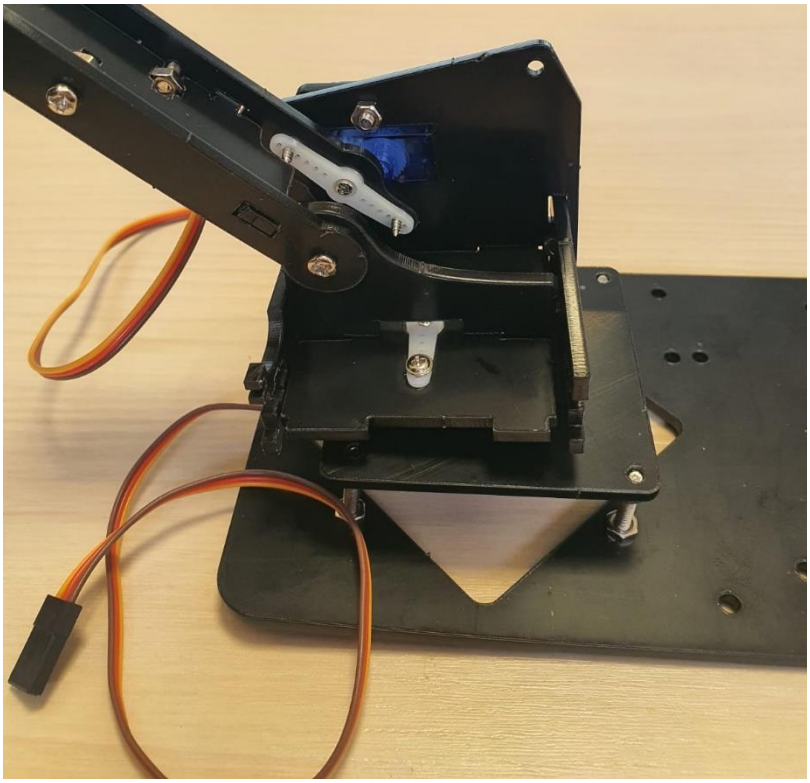






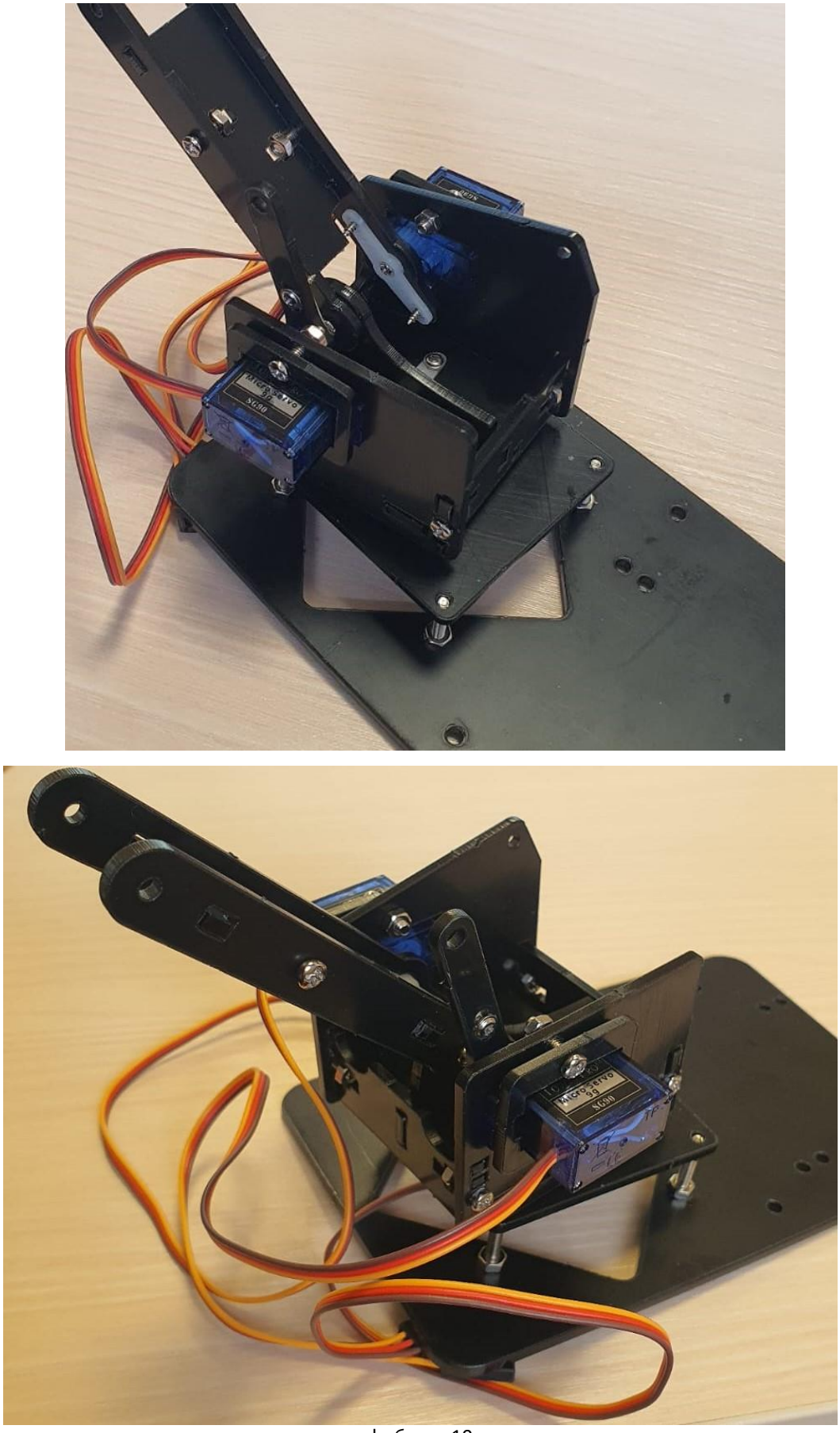






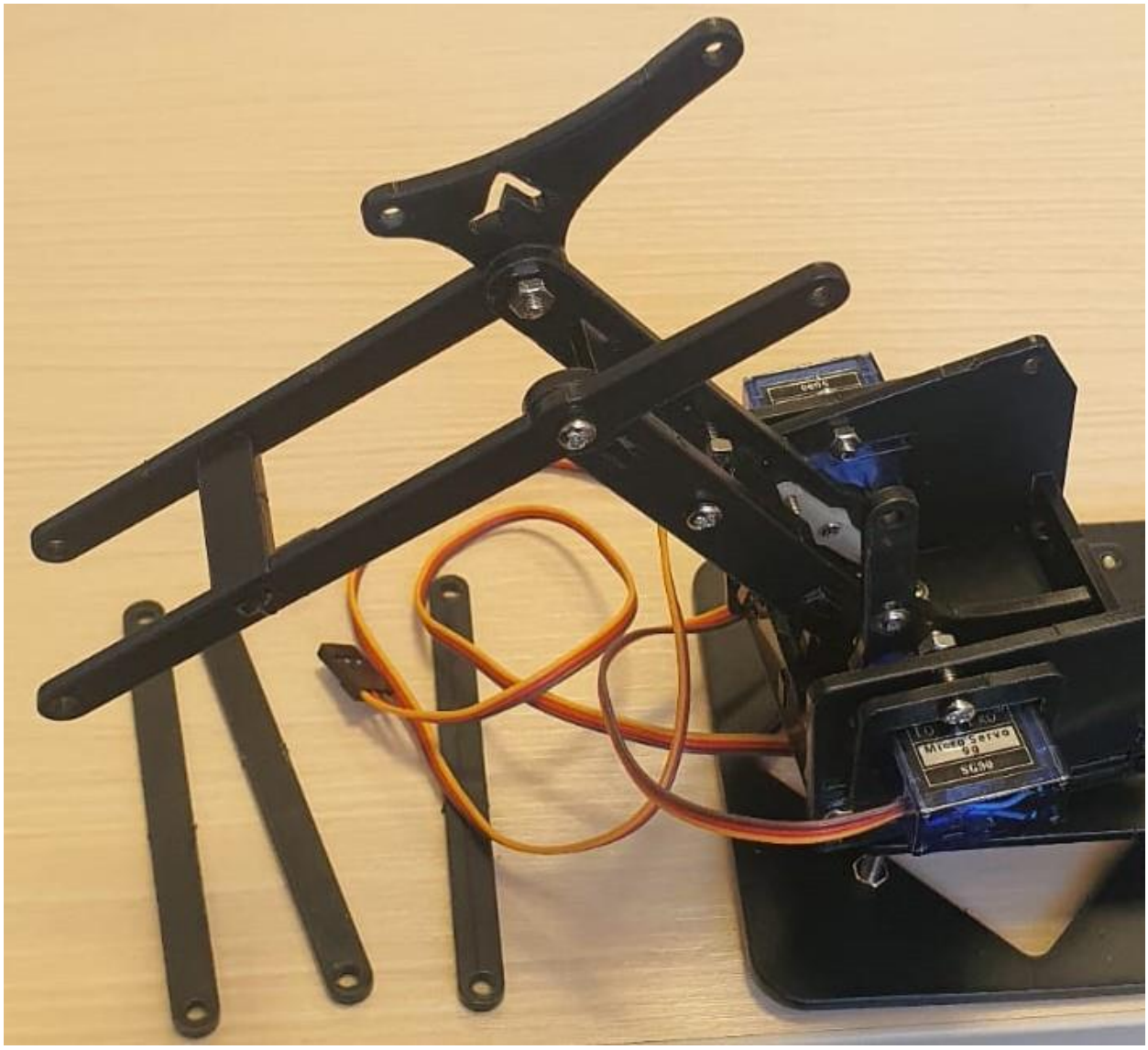
სურათი 9.

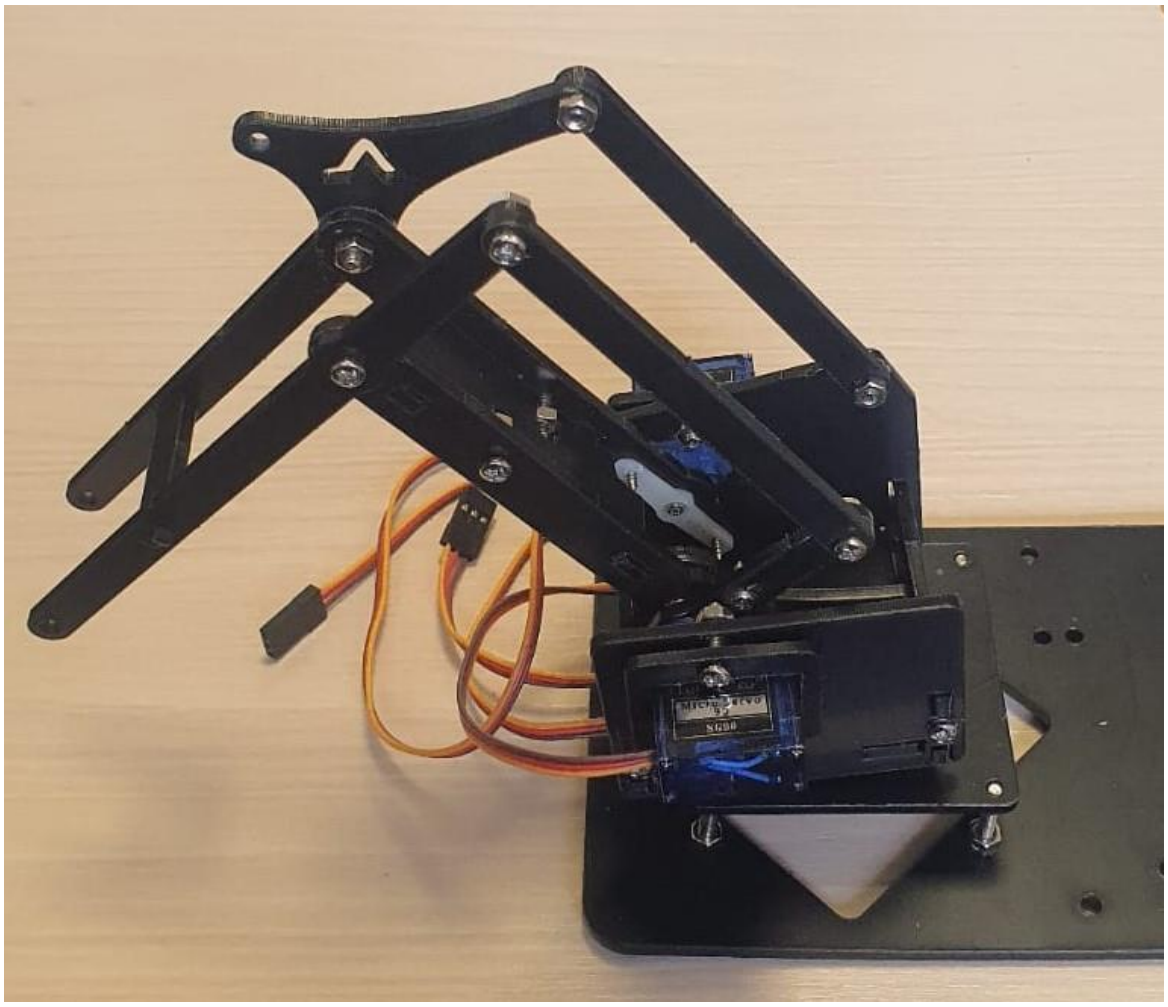
10. დაამაგრეთ მარჯვენა Servo ძრავა ქვედა/საყრდენ კორპუსზე (იხ. სურათი 10):

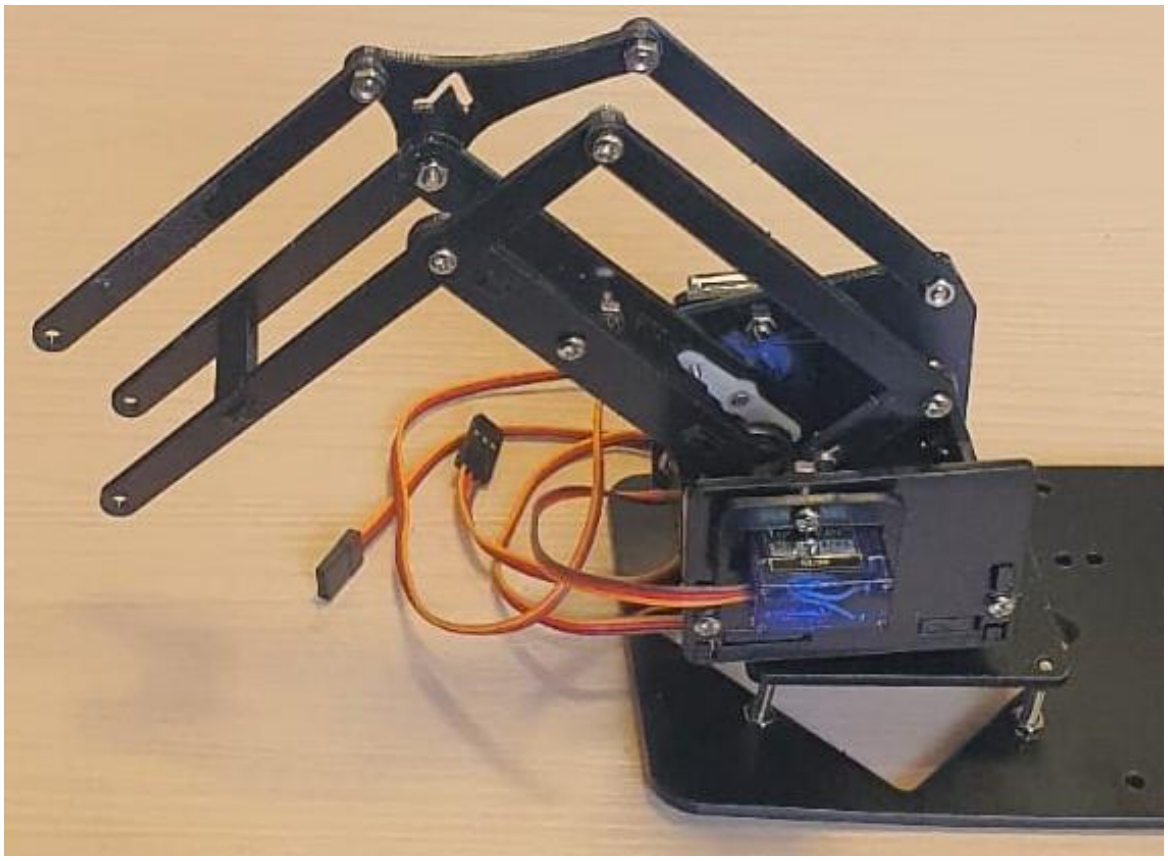


სურათი 10.

11. ახლა დააკავშირეთ მექანიკური მკლავის „იდაყვის სახსრის“ დეტალები (იხ. სურათი 11):

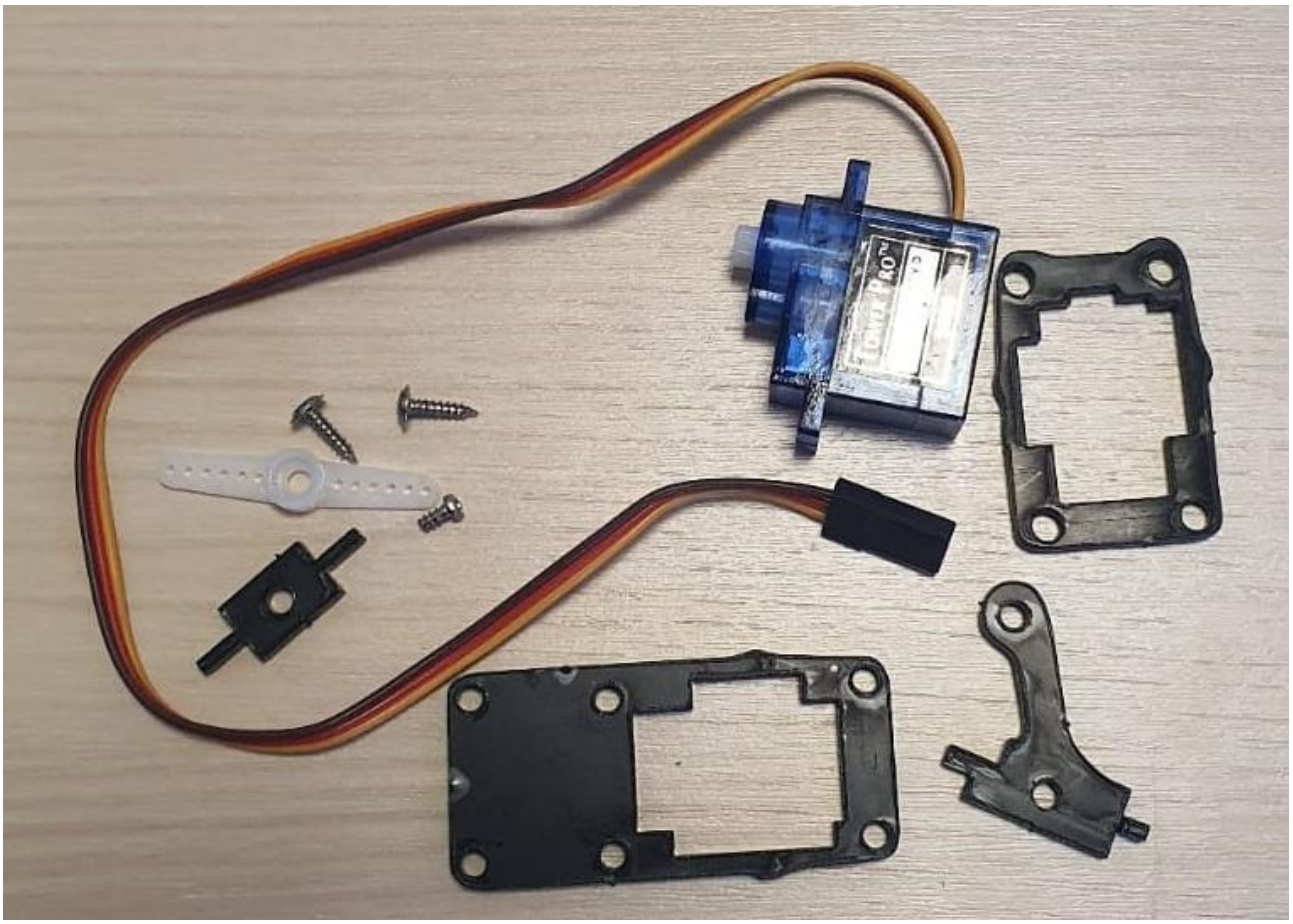


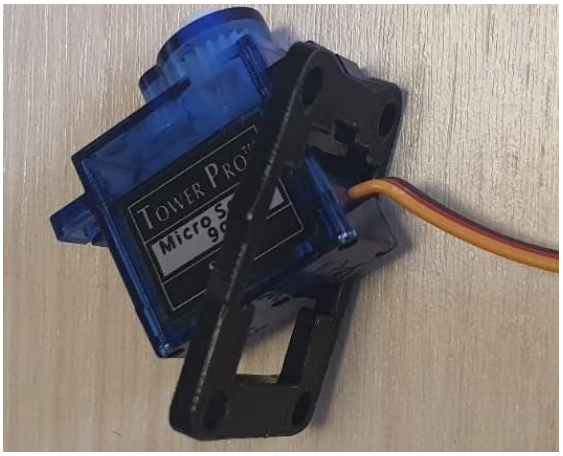


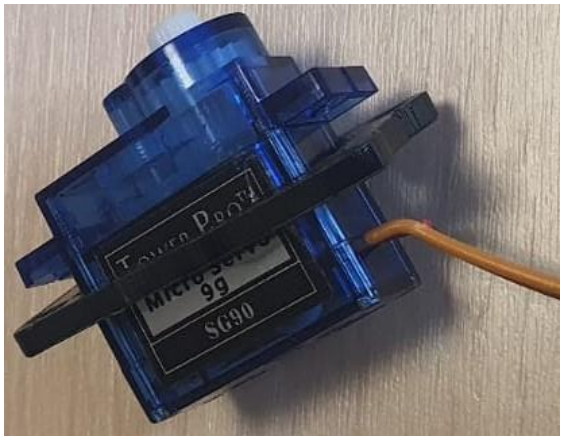


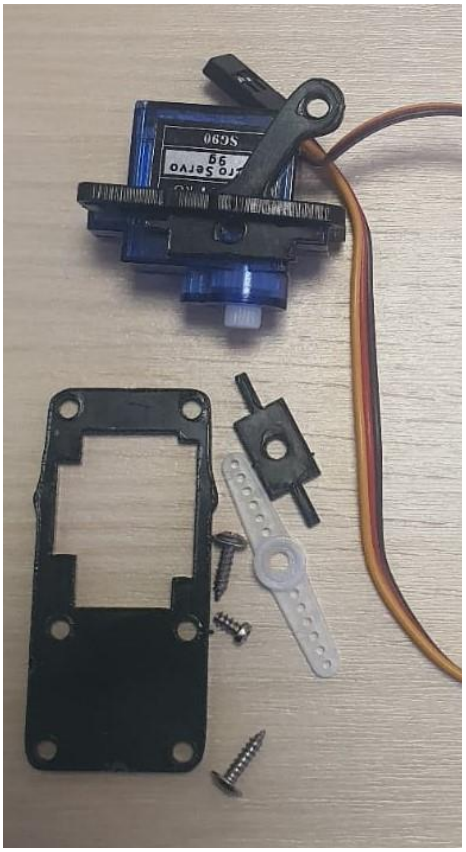
სურათი 11.

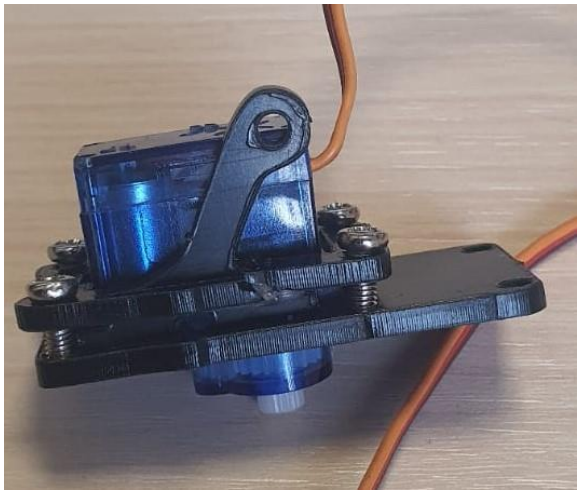
12. ავაწყოთ რობოტი მკლავის „თითები“, სადაც გამოვიყენებთ, ბოლო, Servo4-ს (იხ. სურათი 12):





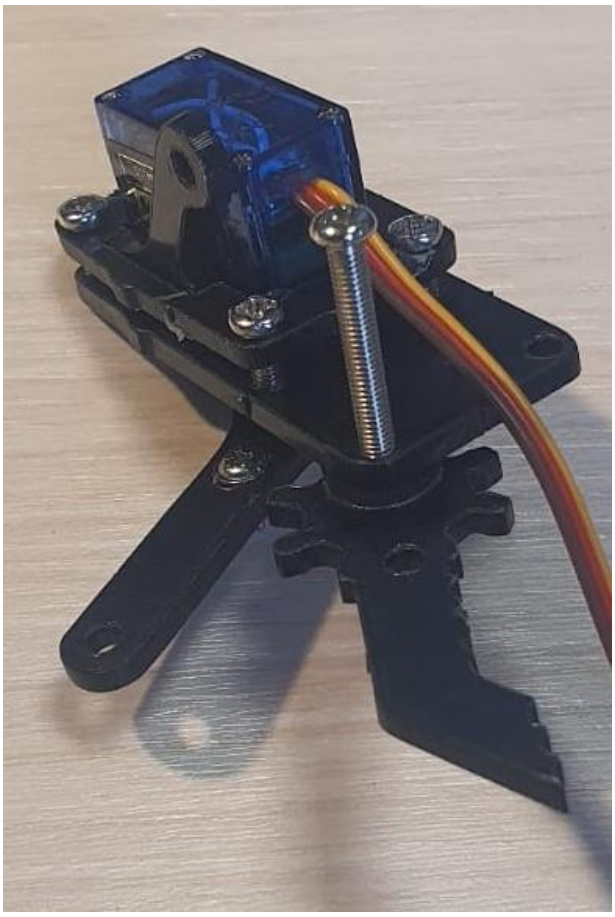


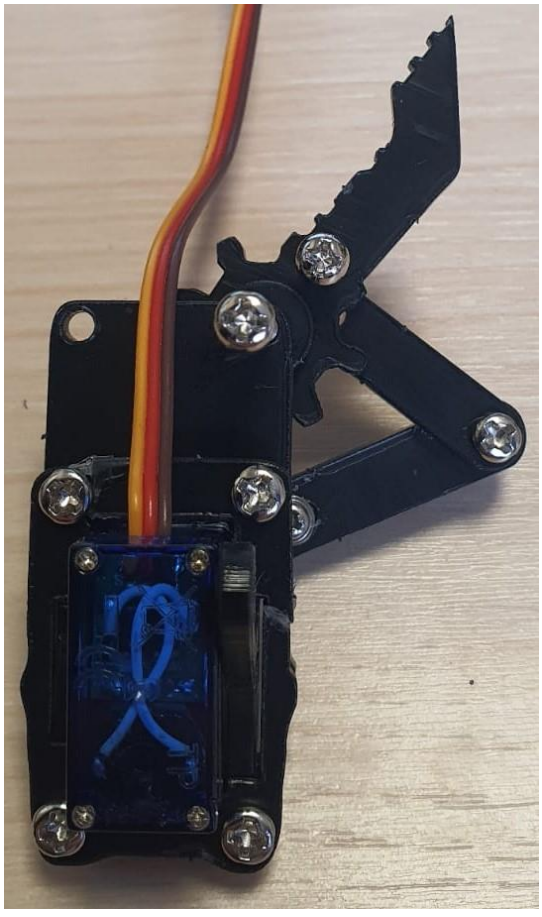


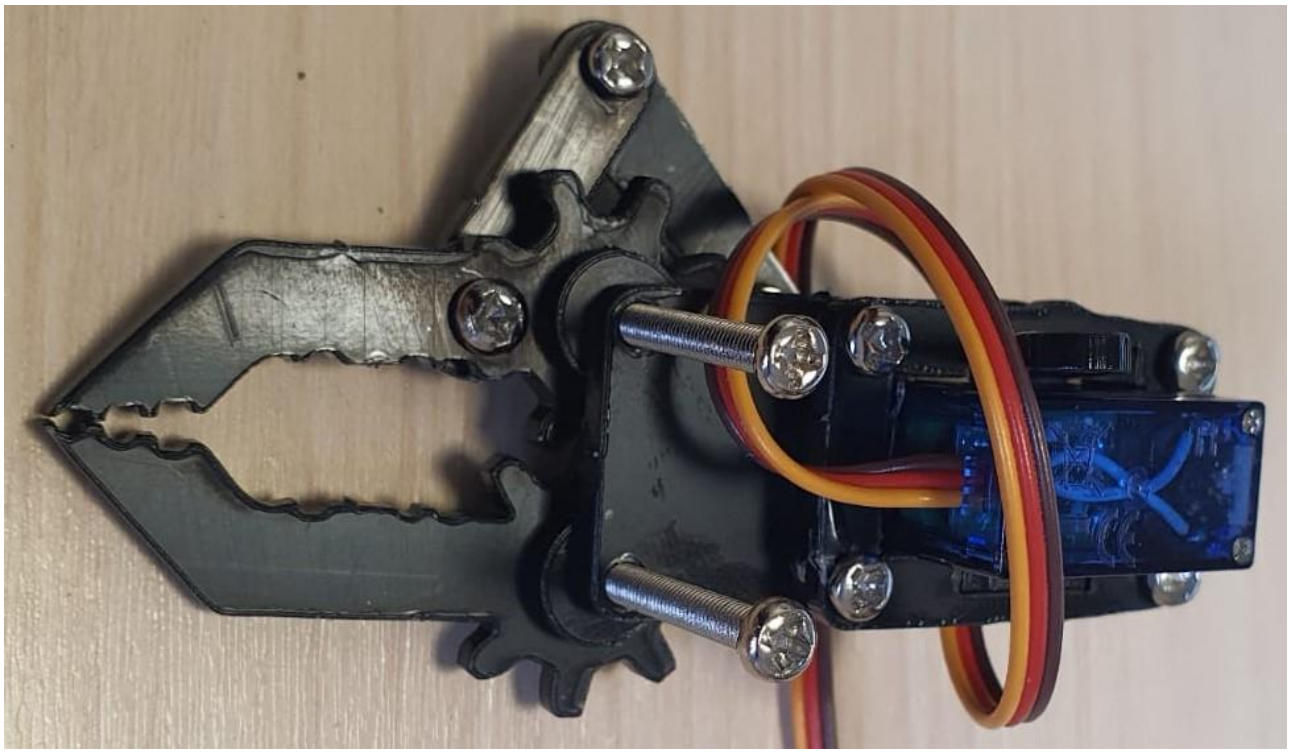


სურათი 12.

13. ახლა დააკავშირეთ Servo ძრავას მბრუნავი ნაწილი „თითების“ მბრუნავ ბერკეტს და ორივე „თითს“ (იხ. სურათი 13):

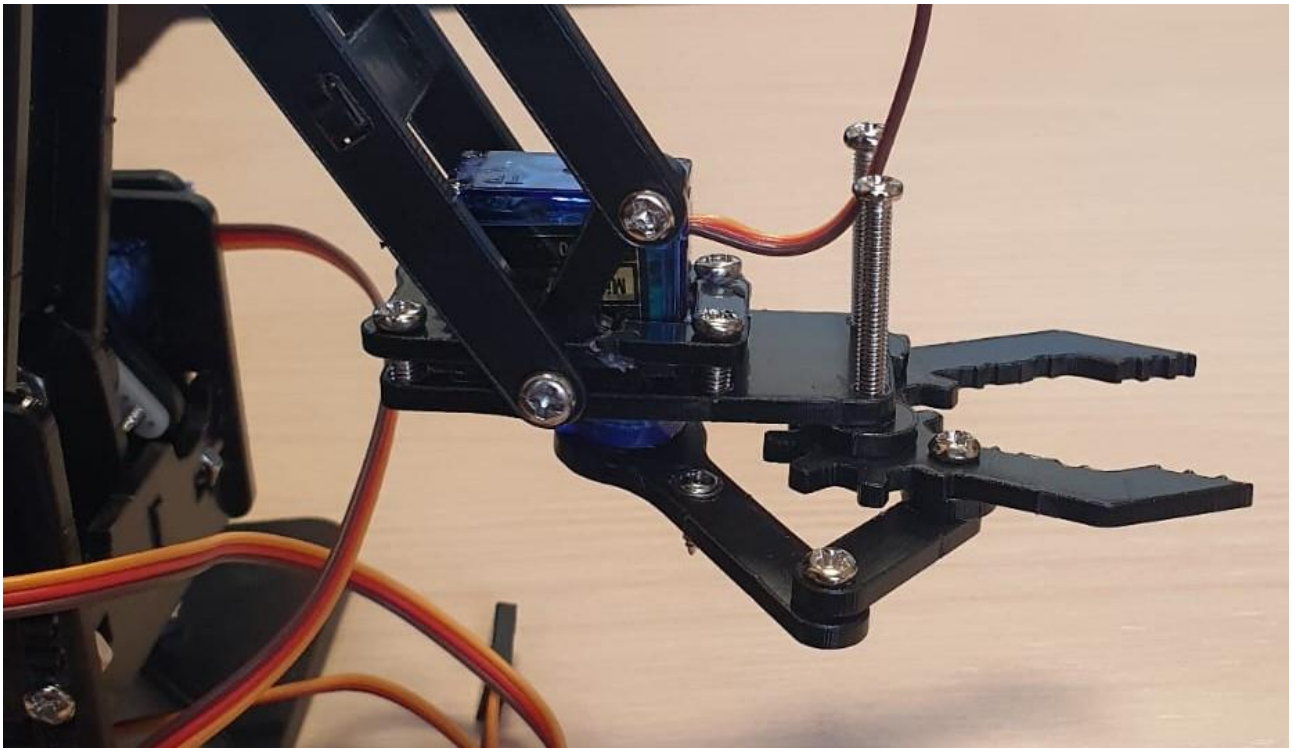


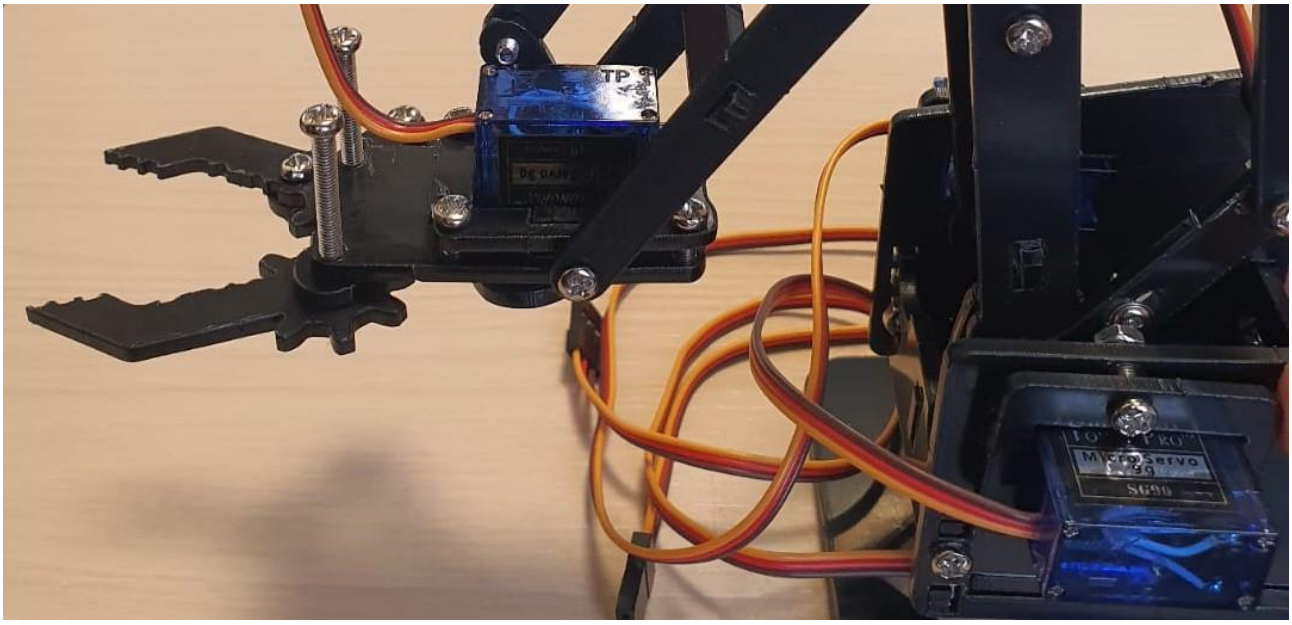


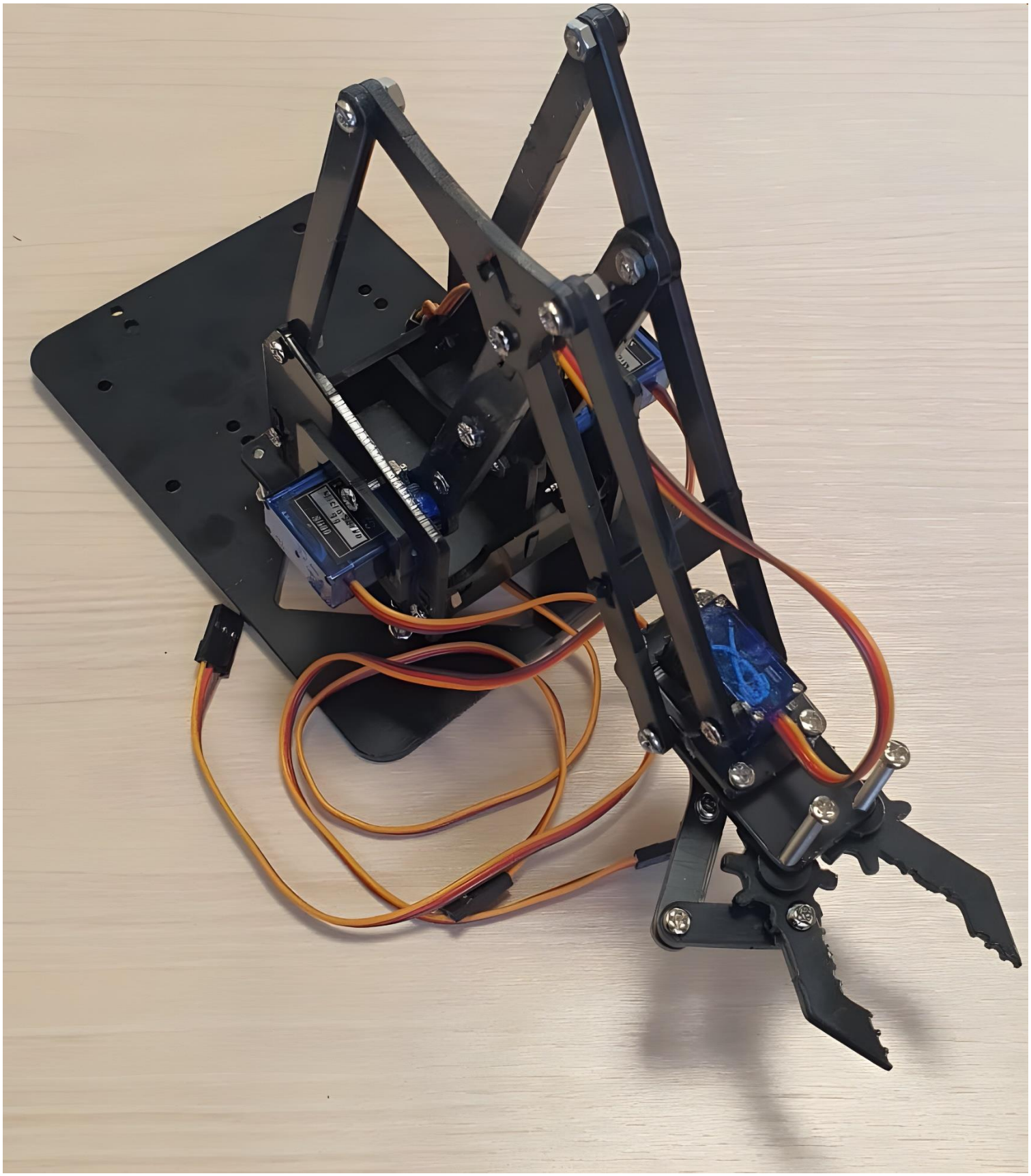


სურათი 13.

14. და ბოლოს, „თითების“ მამოძრავებელი Servo4 დაამაგრეთ მექანიკური მკლავის ძირითად კორპუსს (იხ. სურათი 14):

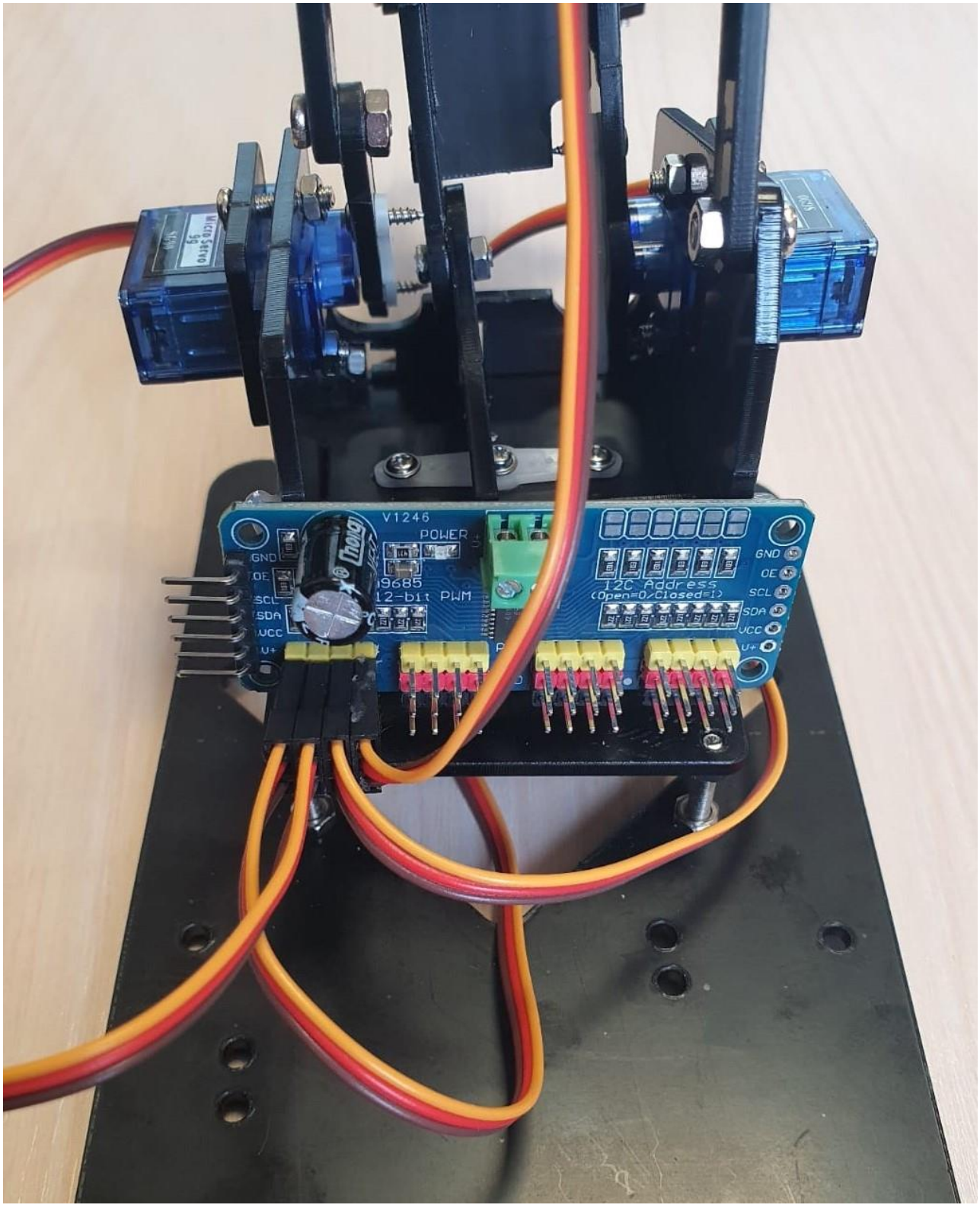






სურათი 14

15. მაშ ასე, რობოტი მკლავი მზად არის. ახლა დაამაგრეთ მასზე PCA9685 PWM Servod რავას დრაივერი და დააკავშირეთ მასთან ოთხივე Servo ძრავა (იხ. სურათი 15 და სურათი 16 -სქემა):



სურათი 15.

ავაწყოთ სქემა

1. დააკავშირეთ არდუინო და PCA9685 PWM Servo ძრავას მოდული:

PCA9685 PWM	Arduino UNO
GND	Pin D9
OE	არ ვიყენებთ
SCL	Pin A5
SDA	Pin A4
VCC	+5V
V+ (მხოლოდ საჭიროების შემთხვევაში)	VIN

2. დააკავშირეთ ერთმანეთთან 4 ცალი Servo ძრავა PCA9685 PWM მოდულთან:

PCA9685 PWM მოდულის პორტები	Servo ძრავა
პორტი 0	Servo 1
პორტი 1	Servo 2
პორტი 2	Servo 3
პორტი 3	Servo 4

16. დააკავშირეთ არდუინო კომპიუტერთან.

შეხსენება!

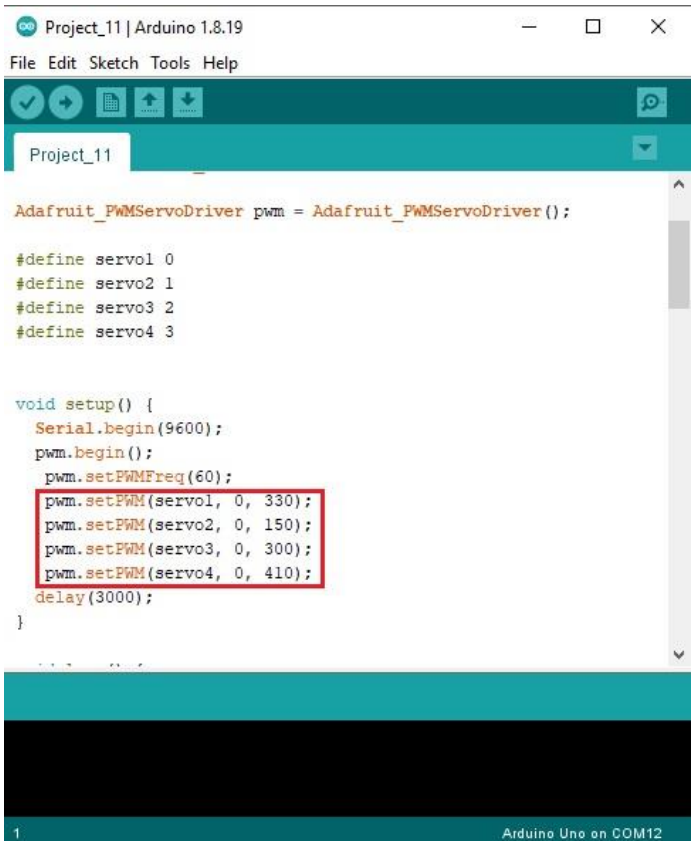
კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools

და მონიშნეთ Arduino UNO და პორტი.

17. ჩატვირთეთ კოდი (Project_11) და ისიამოვნეთ პროექტით.

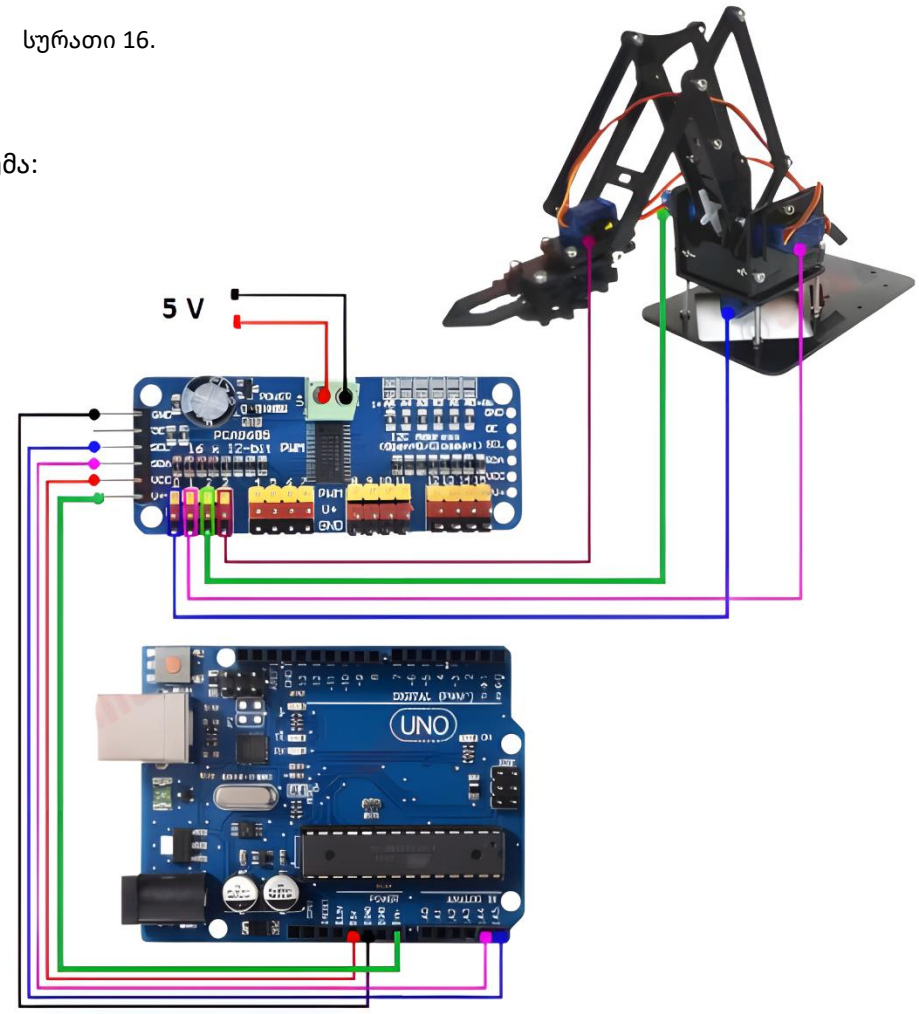
***გაითვალისწინეთ**, რომ თქვენი და კოდში არსებული Servo ძრავების საწყისი პოზიციები, დიდი ალბათობით, განსხვავებული იქნება. ამიტომ, თქვენ დაგჭირდებათ თქვენი Servo ძრავების პოზიციების კალიბრაცია (იხ. X პროექტი, გვ. 117) - ეს უფრო მარტივი მეთოდია* - კოდი Project_11-ის მიხედვით (იხ. სურათი 16). თუ არა და მოგიწევთ *კოდში შესაბამისი ცვლილებების შეტანა.*

ეს არის კოდის მონაკვეთი, სადაც განსაზღვრულია Servo ძრავების საწყისი პოზიციები:



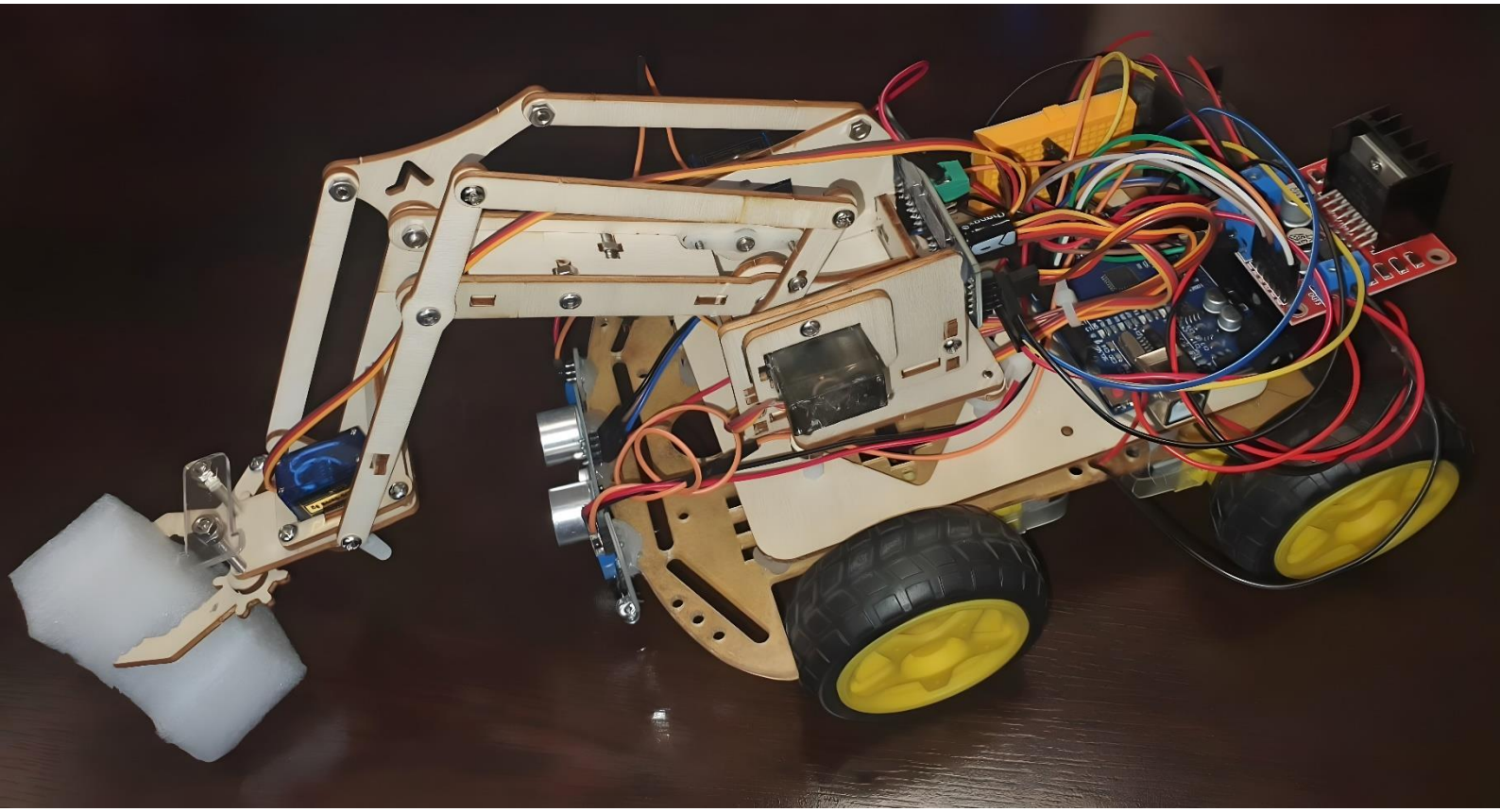
სურათი 16.

ასე გამოიყურება სქემა:



სურათი 17. სქემა

მანქანა რობოტი მკლავით



საჭირო რესურსი

- Arduino Uno
- 4WD მანქანის კომპლექტი
- ძრავას დრაივერი L298N
- PCA9685 PWM Servo ძრავას დრაივერი (16 არხიანი 12-ბიტიანი)
- 4 ცალი Tower Pro SG90 9g
- 2 ცალი ინფრაწითელი სენსორი
- ულტრაბგერითი სენსორი
- რობოტი მკლავი
- 2 ცალი ლითიუმის ელემენტი და ბუდე

- 3 ცალი 1.5V ბატარეა და ბუდე
- პატარა სამაკეტო დაფა
- მამალ-მამალი და დედალ-მამალი გამტარები

საჭირო ბიბლიოთეკა

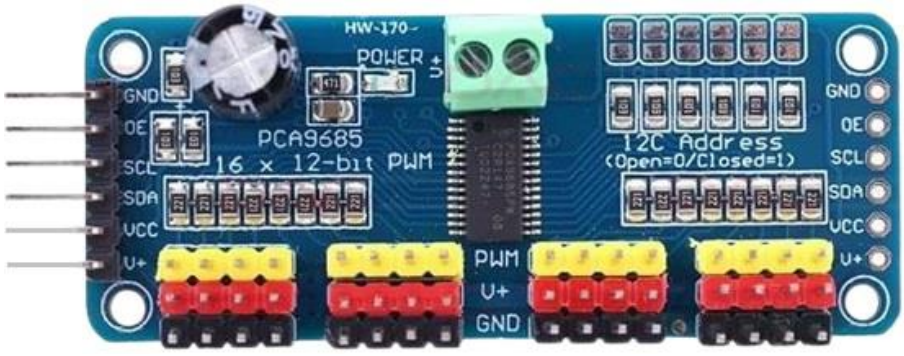
- PWMServoDriver
- Wire
- Tone
- NewPing

როგორ მუშაობს პროექტი

ამ პროექტში ჩვენ გამოვიყენებთ თქვენთვის უკვე ნაცნობ (იხ. პროექტი X და XI) კომპონენტებს: PWM Servo ძრავას დრაივერსა და რობოტ მკლავს. მათი დახმარებით ჩვენ შევქმნით რობოტ მანქანას, რომელიც იმოძრავებს შავ ზოლზე, დააფიქსირებს დამაბრკოლებელ ობიექტს, რობოტი მკლავის საშუალებით აიღებს, გადადებს გვერდზე და გააგრძელებს მოძრაობას.

მოკლედ გავიხსენოთ PWM Servo ძრავას დრაივერის მუშაობის პრინციპები.

PWM Servo ძრავას დრაივერი არის სპეციალიზებული დრაივერი, რომელიც განკუთვნილია **PWM (Pulse Width Modulation)** Servo ძრავების მართვისთვის (იხ. სურათი 1).



სურათი 1. PWM Servo ძრავას დრაივერი

როგორ მუშაობს PWM Servo ძრავას დრაივერი:

1. **PWM სიგნალის მიღება** - იღებს სტანდარტულ PWM სიგნალს მიკროკონტროლერიდან (1-2ms იმპულსები, 20ms პერიოდით).
2. **სიმძლავრის გაზრდა** - Servo ძრავას უზრუნველყოფს საჭირო სიმძლავრით, რასაც მიკროკონტროლერი ვერ უზრუნველყოფს.
3. **პოზიციის კონტროლი** - PWM სიგნალის სიგანის მიხედვით აკონტროლებს Servo ძრავას ლილვის პოზიციას.

ძირითადი მახასიათებლები:

- მართავს მრავალ Servo ძრავას ერთდროულად
- ცალკე ენერგიის წყარო Servo ძრავებისთვის (5-6V)
- იცავს მიკროკონტროლერს გადატვირთვისგან
- მარტივი ინტერფეისი: Signal, V+, GND

გამოყენება: რობოტოტექნიკა, RC მოდელები, კამერების სტაბილიზატორები, სისტემები სწორი პოზიციონირებისთვის.

ავაწყოთ სქემა

1. დააკავშირეთ ერთმანეთთან არდუინო და ძრავას დრაივერი:

ძრავას დრაივერი L298N	Arduino UNO
ENA	Pin D9
IN1	Pin D2
IN2	Pin D3
IN3	Pin D4
IN4	Pin D5
ENB	Pin D10

2. დააკავშირეთ ერთმანეთთან არდუინო და ულტრაბგერითი სენსორი:

ულტრაბგერითი სენსორი	Arduino UNO
VCC (5V)	5V
Trig	Pin A1
Echo	Pin A0
GND	GND

3. დააკავშირეთ ერთმანეთთან არდუინო და ინფრაწითელი სენსორები:

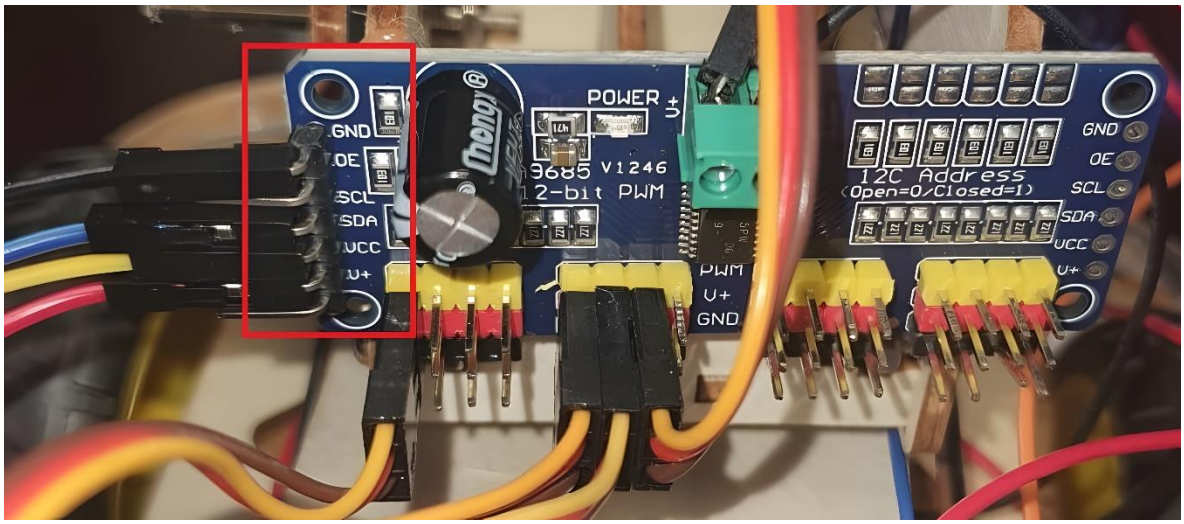
ინფრაწითელი (IR) სენსორი		Arduino UNO
ორივე სენსორის	GND	GND
ორივე სენსორის	5V	5V
მარჯვენა სენსორის	Out	Pin A3
მარცხენა სენსორის	Out	Pin A2

4. დააკავშირეთ ერთმანეთთან არდუინო და პიეზო დინამიკი:

Piezo	Arduino UNO
GND	GND
5V	Pin D7

5. დააკავშირეთ ერთმანეთთან არდუინო და PWM Servo ძრავას დრაივერი (იხ. სურათი 2):

PWM Servodრავას დრაივერი	Arduino UNO
GND	GND
SCL	Pin A5
SDA	Pin A4
VCC	5V

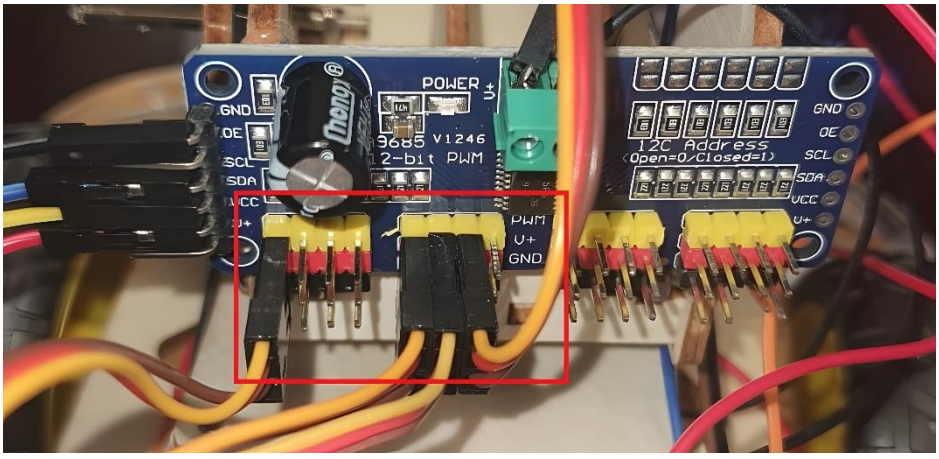


სურათი 2. არდუინო და PWM Servo ძრავას დრაივერის კავშირები

6. დააკავშირეთ ერთმანეთთან Servo ძრავები და PWM Servo ძრავას დრაივერი:

Servo	PWM Servoძრავას დრაივერი
Servo 1	პორტი 0
Servo 2	პორტი 4
Servo 3	პორტი 5
Servo 4	პორტი 6

PWM Servo ძრავას დრაივერის პორტები შეგიძლიათ თქვენი სურვილით აირჩიოთ. ამ შემთხვევაში კოდშიც უნდა შეიტანოთ ცვლილება და თქვენი არჩეული პორტები მონიშნოთ. ჩვენი არჩევანის მიზეზი გახდა პირველი, მეორე და მესამე პორტების დაზიანება (იხ. სურათი 3).



სურათი 3. PWM Servod რავას დრაივერისა და Servo ძრავების კავშირები

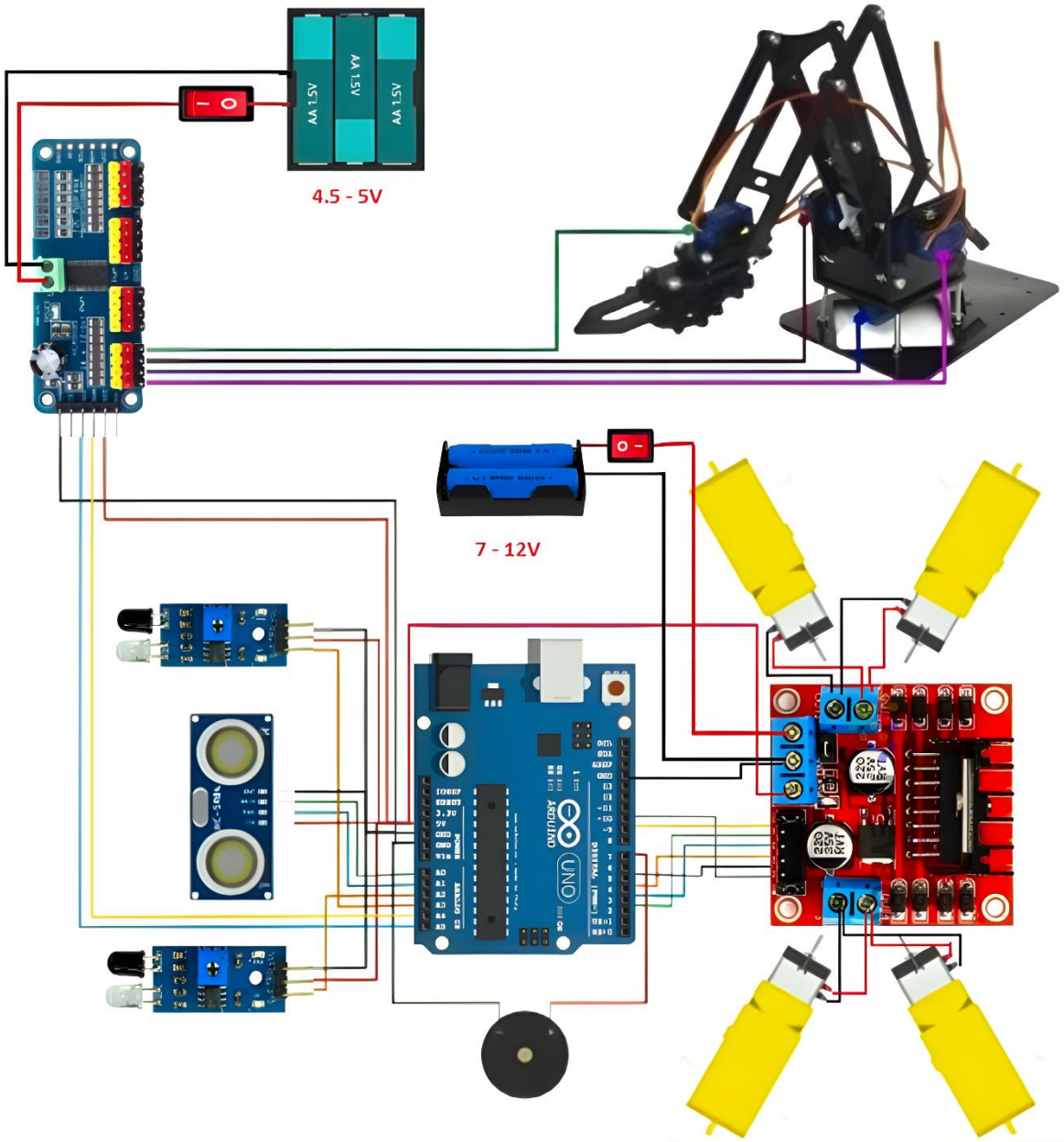
ეს მნიშვნელოვანია!

პროექტში გამოყენებული სენსორების რაოდენობიდან გამომდინარე, არდუინოს 5V-ისა და GND-ს პინები არ არის საკმარისი. ამ კავშირებისთვის ვიყენებთ სამაკეტო დაფას (ძირითადი სქემა იხ. გვ. 17, სურათი 11 და დაამატეთ დანარჩენი სენსორები).

7. სამაკეტო დაფაზე დააკავშირეთ ერთმანეთთან არდუინო და სენსორები: ძრავას დრაივერი, PWM Servo ძრავას დრაივერი, ულტრაბგერითი და ინფრაწითელი სენსორები (ყურადღება მიაქციეთ ცხრილის ფერებს, ისინი დაწყვილებულია, მაგალითად, არდუინო-სამაკეტო დაფა: არდუინოს 5V უკავშირდება დაფის დადებითპოლუსიან (+) ლიანდაგს, ხოლო არდუინოს GND - დაფის უარყოფითპოლუსიან ლიანდაგს (GND, -)).

Arduino UNO	სამაკ. დაფა	ძრავას დრაივერი L298N	სამაკ. დაფა	PWM Servoძრავას დრაივერი	სამაკ. დაფა	ულტრაბგერის სენსორი	სამაკ. დაფა	IR სენსორი	სამაკ. დაფა
5V	+	+5V	+	V+	+	VCC	+	VCC	+
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND

პროექტის სქემა ასე გამოიყურება (იხ. სურათი 4):



სურათი 4. სქემა

5. დააკავშირეთ არდუინო კომპიუტერთან

შეხსენება!
კოდის ჩატვირთვამდე არდუინოს IDE ინტერფეისის მენიუს ზოლში გახსენით Tools
და მონიშნეთ Arduino UNO და პორტი.

6. ჩატვირთეთ კოდი.

გამორთეთ არდუინო კომპიუტერიდან. ჩაალაგეთ ელემენტები ბუდეში და მართეთ მანქანა.

როგორ მუშაობს კოდი

ეს კოდი აკონტროლებს 4WD მანქანას რობოტი მკლავით, რომელიც აერთიანებს ხაზის მიმდევარის, დაბრკოლების დაძლევისა და ავტონომიური ნავიგაციის ფუნქციებს.

- **ძირითადი კომპონენტები:**
  - **4WD წამყვანი სისტემა** - 4 DC ძრავა H-ხიდის მართვით
  - **რობოტი მკლავი** - 4 Servo ძრავით PWM დრაივერის მეშვეობით

- **სენსორები**:
  - ულტრაბგერითი სენსორი (დაბრკოლების აღმოსაჩენად)
  - 2 ინფრაწითელი სენსორი (ხაზის მიმდევრისთვის)
  - ბაზერი (ხმოვანი სიგნალებისთვის)

მუშაობის ძირითადი პრინციპი:

1. **ინიციალიზაცია (setup()):**
   - ახდენს ყველა პინის კონფიგურაციას
   - Servo ძრავებს აყენებს საწყის ("home") პოზიციაზე
   - ასრულებს გაშვების ხმის სიგნალს

2. **მთავარი ციკლი (loop()):**

- **დაბრკოლების შემოწმება**

ულტრაბგერითი სენსორით ზომავს მანძილს, თუ დაბრკოლება 2-15 სმ დიაპაზონშია:

- გააჩერებს ძრავებს
- დაელოდება 1 წამს (WAIT_BEFORE_ARM)
- გააააქტიურებს რობოტ მკლავს

- **რობოტი მკლავის მუშაობა (optimizedArmSequence())**

როდესაც დაბრკოლებას აღმოაჩენს, მკლავი ასრულებს 10 ეტაპიან მოქმედებას:

1. დახრა ობიექტისკენ
2. “თითების” გახსნა
3. აწევა
4. გაწევა წინ
5. ობიექტის აღება (“თითების” დახურვა)
6. მაღლა აწევა
7. ბრუნვა
8. დაწევა დასაყენებლად
9. ობიექტის გათავისუფლება
10. საწყის პოზიციაზე დაბრუნება

- **ხაზის მიმდევარი რეჟიმი (linefollower())**

როცა დაბრკოლება არ არის ან მკლავი არ მუშაობს:

ორი ინფრაწითელი (IR) სენსორის მიხედვით მიჰყვება ხაზს:

- ორივე სენსორზე თეთრი → წინ
- ორივე სენსორზე შავი → გაჩერება

- მარცხენა შავი → მარჯვნივ ბრუნი
- მარჯვენა შავი → მარცხნივ ბრუნი

3. **მოძრაობის შემდეგ/დაბრკოლების აღების შემდეგ:**

- რობოტი უკან იხევს 500ms
- შემდეგ წინ მიემართება 800ms
- ხელახლა იწყებს ხაზის მიყოლას

ტექნიკური დეტალები:

- **PWM სიხშირე**: 60Hz Servoებისთვის
- **Servo კონტროლი**: Adafruit PWM Servo Driver
- **ძრავის სიჩქარე**: Speed = 100 (0-255 მასშტაბით)
- **Servo მოძრაობა**: გლუვი, საფეხურებად მართვა (moveServoSmoothly())

უსაფრთხოების მექანიზმები:

- armActivated ცვლადი უზრუნველყოფს, რომ მკლავის მოქმედების დროს სხვა ფუნქციები არ გაეშვას.
- 1 წამის ლოდინი დაბრკოლების აღმოჩენის შემდეგ.
- სისტემატური დებაგინგი* სერიალ პორტზე

*გამართვა (ინგლისურად "debugging") შეცდომების გამოსწორების პროცესს ნიშნავს პროგრამირებაში. ეს არის ტექნიკა, რომლითაც პროგრამისტი პოულობს და აღმოფხვრის ხარვეზებს, "ბაგებს" (bugs) კოდში. პროგრამული უზრუნველყოფის შემთხვევაში, გამართვის ტაქტიკა შეიძლება მოიცავდეს ინტერაქტიულ გამართვას, კონტროლის ნაკადის ანალიზს.

ეს რობოტი არის ავტონომიური სისტემა, რომელსაც შეუძლია ხაზის მიდევნება, დაბრკოლებების აღმოჩენა და დაძლევა, შემდეგ კი ხაზზე მოძრაობის გაგრძელება.

კოდის მუშაობის პრინციპი

🏗️ **არქიტექტურა** (3 ძირითადი ნაწილი):

[INPUT] → [PROCESSING] → [OUTPUT]
სენსორები Arduino ძრავები/მკლავი

🔄 **მთავარი ციკლი (loop() ფუნქცია)**:

void loop() {

1. ⚡ ელექტროენერგიის ჩართვა
2. 🔍 სენსორებიდან მონაცემების წაკითხვა
3. 🧠 მონაცემების დამუშავება და გადაწყვეტილების მიღება
4. ⚙️ ბრძანებები DC ძრავებს/Servo ძრავებს
5. 🔁 პროცესის გამეორება

}

📊 **სენსორული სისტემა** (3 ტიპის სენსორი):

1. ხაზის სენსორები - ინფრაწითელი (IR):

S1 (A2) → მარცხენა ხაზის სენსორი
S2 (A3) → მარჯვენა ხაზის სენსორი

ლოგიკა:

- `0` = ხაზი ჩანს (შავი)
- `1` = ხაზი არ ჩანს (თეთრი)

2. ულტრაბგერითი სენსორი:

Trig (A1) → სიგნალის გაგზავნა
Echo (A0) → სიგნალის მიღება

ფორმულა: `მანძილი = (დრო × ხმის სიჩქარე) / 2`

3. ბაზერი:

BUZZER_PIN (7) → ხმის სიგნალები

🎯 **გადაწყვეტილების მიღების ლოგიკა**:

საგანგებო პრიორიტეტები:

1. დაბრკოლება (15 სმ) → STOP
2. ხაზის მიმდევარი → MOVE

if-else პირობები:

```
if (დაბრკოლება <= 15 სმ) {
   // 1. გაჩერება
   // 2. 1 წამი დაცდა
   // 3. მკლავის აქტივაცია
   // 4. უკან გასვლა
   // 5. წინ გაგრძელება
} else {
   // ხაზის მიმდევარი
}
```

🤖 **რობოტის 4 ძირითადი მდგომარეობა**:

1. მოძრაობის რეჟიმები:

FORWARD: IN1=H, IN2=L | IN3=H, IN4=L

BACKWARD: IN1=L, IN2=H | IN3=L, IN4=H

RIGHT: IN1=H, IN2=L | IN3=L, IN4=H

LEFT: IN1=L, IN2=H | IN3=H, IN4=L

STOP: IN1=L, IN2=L | IN3=L, IN4=L

2. ხაზის მიმდევარის ლოგიკა:

თუ S1=0 და S2=0 → წინ (ორივე ხაზზეა)

თუ S1=1 და S2=0 → მარჯვნივ (მარცხენა გარეთ)

თუ S1=0 და S2=1 → მარცხნივ (მარჯვენა გარეთ)

თუ S1=1 და S2=1 → გაჩერება (ხაზი დაიკარგა)

3. დაბრკოლების რეაგირება:

1. 📏 მანძილის გაზომვა (15 სმ)
2. 🛑 დაუყოვნებლივი გაჩერება
3. ⏱️ 1000ms დაცდა
4. 💪 მკლავის მოქმედება
5. 🔄 უკან გასვლა (500ms)
6. 🚗 წინ გაგრძელება (800ms)

4. მკლავის მოქმედებების თანმიმდევრობა:

1. 📐 დახრა (S1: 220→220)
2. ✋ თითების გაღება (S4: 410→510)
3. ⬆️ აწევა (S2: 100→250)
4. ➡️ გაწევა (S3: 300→450)
5. 🎯 ობიექტის აღება (S4: 510→410)
6. ⬆️ მაღლა აწევა (S2: 100→250)
7. 🔃 ბრუნვა (S1: 220→450)
8. ⬇️ დაწევა (S2: 250→250)
9. 🎯 ობიექტის გათავისუფლება (S4: 410→510)
10. 🏠 საწყის პოზიციაზე დაბრუნება

⚡ **ენერგეტიკული ნაკადი (კვება)**:

[12V Battery] → [L298N Driver] → [4 ძრავა]

↓

[5V Regulator] → [Arduino]

↓

[PCA9685 PWM] → [4 Servo]

↓

[5V/3.3V] → [სენსორები]

🎛️ **კონტროლის იერარქია**:

Arduino Uno (Brain)

├── L298N (4WD ძრავების კონტროლი)

├── PCA9685 (Servoების კონტროლი)

├── Ultrasonic Sensor (დაბრკოლების დეტექტორი)

├── IR Sensors (ხაზის დეტექტორი)

└── Buzzer (ხმის სიგნალები)

🕐 **დროის მენეჯმენტი**:

millis() ფუნქცია:

- ხაზის მიმდევარი: ყოველ 50ms
- დაბრკოლების შემოწმება: ყოველ 300ms
- Serial დებაგინგი: ყოველ 500ms
- მკლავის დაცდა: 1000ms

🔊 **ხმის სიგნალები**:

1000Hz, 200ms → გაშვება

1500Hz, 100ms → დაბრკოლების აღმოჩენა

1200Hz, 200ms → ობიექტის აღება

1000Hz, 200ms → ობიექტის გათავისუფლება

2000Hz, 200ms → დავალების შესრულება

🔧 **პარალელური დამუშავება**:

[Thread 1]: ხაზის მიმდევარი (50ms ინტერვალი)

[Thread 2]: დაბრკოლების მონიტორინგი (უწყვეტი)

[Thread 3]: მკლავის კონტროლი (როცა აქტიურია)

[Thread 4]: ხმის სიგნალები (საჭიროებისამებრ)

🎯 **მიზანი და ფუნქციონალი**:

მიზანი: "მიჰყვე ხაზს, იპოვე დაბრკოლება, გადაიტანე და გააგრძელე გზა"

ფუნქციონალი:

1. 🧍 მიჰყვება შავ ხაზს
2. 👁 ხედავს დაბრკოლებას 15 სმ დისტანციიდან
3. 🛑 ჩერდება და 1 წამს ელოდება
4. 🦾 ასრულებს მკლავის მოქმედებას
5. 🔁 ოდნავ უკან იხევს
6. 🚗 აგრძელებს ხაზის მიყოლას

☑ **სისტემის ევოლუცია**:

1.0: მხოლოდ ხაზის მიმდევარი

2.0: + დაბრკოლების დეტექტორი

3.0: + რობოტი მკლავი

4.0: + ბაზერი (ხმის სიგნალები)

5.0: + დროის მენეჯმენტი

6.0: + გაუმჯობესებული ალგორითმი

ეს არის სრულფასოვანი ავტონომიური სისტემა, რომელიც ინტეგრირებს სენსორულ აღქმას, გადაწყვეტილების მიღებასა და მექანიკურ მოქმედებას ერთ სისტემაში!

რა მნიშვნელობა აქვს ასეთ მანქანებს?!

🎓 საგანმანათლებლო/სასწავლო მნიშვნელობა:

1. **STEM განათლება**:
    - **S**cience (მეცნიერება) - ფიზიკა, მათემატიკა
    - **T**echnology (ტექნოლოგია) - პროგრამირება, ელექტრონიკა
    - **E**ngineering (ინჟინერია) - მექანიკა, დიზაინი
    - **M**athematics (მათემატიკა) - ალგორითმები, გამოთვლები
2. **პრაქტიკული უნარები**:
    - პრობლემის გადაჭრა
    - კრიტიკული აზროვნება
    - პროექტის მართვა
    - გუნდური მუშაობა
3. **ტექნიკური უნარები**:
    - პროგრამირება (Arduino, C++)
    - ელექტრონიკა
    - მექანიკა
    - სენსორების გამოყენება

🏭 პრაქტიკული/რეალური გამოყენება:

1. **საწარმოო/ლოჯისტიკა**:
    - საწყობის რობოტები (Amazon, Alibaba)
    - ავტომატური შესაფუთი სისტემები
    - მასალების გადატანა
2. **სოფლის მეურნეობა**:
    - ავტონომიური ტრაქტორები
    - მოსავლის აღმწერი რობოტები
    - ავტომატური სარწყავი სისტემები
3. **სამედიცინო**:
    - წამლების გადაცემის რობოტები საავადმყოფოებში
    - დისტანციური ქირურგია
    - რეაბილიტაციის დამხმარე მოწყობილობები

4. **გარემოს დაცვა**:
    - ნაგვის შემგროვებელი რობოტები
    - გაწმენდის რობოტები
    - მონიტორინგის სისტემები

🚗 **ავტომობილების ინდუსტრია**:

1. **ავტონომიური მანქანები** - თქვენი პროექტი არის **მინიატურული ვერსია**:
    - Tesla Autopilot
    - Waymo (Google)
    - Cruise (GM)
2. **ADAS სისტემები** (Advanced Driver Assistance Systems):
    - ავტომატური გაჩერება
    - ხაზის მიმდევარი
    - შეჯახების თავიდან აცილება

🏆 **თქვენი პროექტის კონკრეტული მნიშვნელობა**:

1. **ავტონომიური მოძრაობის საფუძვლები**:
    - სენსორული აღქმა (ულტრაბგერითი, IR)
    - გადაწყვეტილების მიღება
    - მექანიკური მოქმედება (მკლავი)
2. **რეალური პრობლემის გადაჭრა**:
    - "ხედავს" დაბრკოლებას
    - იღებს გადაწყვეტილებას
    - ასრულებს დავალებას
    - აგრძელებს გზას
3. **ინტერდისციპლინარული ინტეგრაცია**:
    - მექანიკა (მანქანა + მკლავი)
    - ელექტრონიკა (სენსორები, ძრავები)
    - პროგრამირება (ლოგიკა, ალგორითმები)

🌍 **დიდი სურათი**:

თქვენი პროექტი არის **ავტონომიური სისტემების პროტოტიპი**, რომელიც:

- **აწვდის მონაცემებს** (სენსორები)
- **აანალიზებს** (Arduino)
- **იღებს გადაწყვეტილებას** (კოდი)
- **მოქმედებს** (ძრავები, მკლავი)

💡 **რატომ არის მნიშვნელოვანი**:

1. **მომავლის ტექნოლოგიები** - ავტონომიური სისტემები ყველგან იქნება
2. **კარიერის გზა** - რობოტოტექნიკა, AI, ინჟინერია
3. **პრობლემის გადაჭრის უნარი** - ყველაზე მნიშვნელოვანი უნარი
4. **კრეატიულობა** - შექმნით იმას, რაც ადრე არ არსებობდა

თქვენი პროექტი არის პატარა საფეხური დიდი სამყაროსკენ: ავტონომიური მანქანები, სმარტ ქალაქები, ინდუსტრია 4.0, რობოტები, რომლებიც გვეხმარებიან ყოველდღიურ ცხოვრებაში!

თქვენ აშენებთ არა მხოლოდ რობოტს, არამედ - მომავალს!

გამოყენებული წყაროები:

1. ზ. ტაბატაძე, თ. თოდუა - არდუინო. პრაქტიკული სახელმძღვანელო დამწყებთათვის. თბილისი, 2019

2. ჯ. გრიგალაშვილი - Arduino-ს ვიზუალური დაპროგრამება FLProg გარემოში. საქართველოს ტექნიკური უნივერსიტეტი. თბილისი, 2015

3. http://Arduino.cc

4. https://Github.com

5. https://srituhobby.com

6. AI პლატფორმები: Claude, Gemini, Deep Seek