

Multiplatformski robot za unapređenje nastave informatike u osnovnim školama

Hadži-Veljković, Nikola

Graduate thesis / Diplomski rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Applied Sciences in Information Technology / Veleučilište suvremenih informacijskih tehnologija**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:289:842216>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-22**

Repository / Repozitorij:

[VSITE Repository - Repozitorij završnih i diplomskih radova VSITE-a](#)



VELEUČILIŠTE SUVREMENIH INFORMACIJSKIH TEHNOLOGIJA
STRUČNI DIPLOMSKI STUDIJ INFORMACIJSKIH TEHNOLOGIJA

Nikola Hadži-Veljković

DIPLOMSKI RAD

**MULTIPLATFORMSKI ROBOT ZA UNAPREĐENJE NASTAVE
INFORMATIKE U OSNOVNIM ŠKOLAMA**

Zagreb, siječnja 2025.

Studij: Stručni diplomski studij informacijskih tehnologija
smjer računalni sustavi
Student: **Nikola Hadži-Veljković**
Matični broj: 2021108

Zadatak diplomskog rada

Predmet: Programsko inženjerstvo
Naslov: **Multiplatformski robot za unapređenje nastave informatike u osnovnim školama**
Zadatak: Ukratko opisati Microbit i Arduino platforme. U praktičnom dijelu izraditi daljinski kontroliran robot koji će za upravljanje moći koristiti dvije različite platforme: Microbit i Arduino. Robot izraditi na način da se može koristiti jedna ili druga platforma, a da se zamjenom platforme ne gubi na funkcionalnosti. Logiku programirati u programskom jeziku Python.
Mentor: Tomislav Soldo, pred.
Zadatak uručen kandidatu: 2.10.2024.
Rok za predaju rada: 30.9.2025.
Rad predan: _____

Povjerenstvo:

Jurica Đurić, v. pred.	član predsjednik	_____
Tomislav Soldo, pred.	mentor	_____
Zoran Radek, pred.	član	_____

SADRŽAJ

1. UVOD.....	7
2. MIKROKONTROLERI.....	9
2.1. Općenito o mikrokontrolerima.....	9
2.2. Osnovna struktura mikrokontrolera.....	9
2.3. Microbit.....	9
2.4. Arduino.....	11
3. KOMPONENTE MULTIPLATFORMSKOG ROBOTA.....	13
3.1. Glavni dijelovi robota.....	13
3.1.1. Senzori prepreka.....	13
4. PRAKTIČNI RAD – MULTIPLATFORMSKI ROBOT ZA UNAPREĐENJE NASTAVE INFORMATIKE U OSNOVNIM ŠKOLAMA.....	15
4.1. L298N Dual H Most.....	15
4.1.1. Upravljanje motorima sa Microbitom.....	15
4.1.2. Upravljanje motorima s Arduino.....	18
4.1.3. Analiza razlike koda pokretanja motora za Microbit i Arduino.....	19
4.2. Ultrazvučni daljinomjer HC-SR04.....	20
4.2.1. Korištenje senzora na Microbitu.....	20
4.2.2. Korištenje senzora na Arduino.....	22
4.3. Izrada platforme robota za više mikrokontrolera.....	24
4.4. Pokretanje s Arduino.....	28
4.5. Sastavljanje završnog multiplatformskog robota.....	32
5. ZAKLJUČAK.....	36
LITERATURA.....	38
SAŽETAK.....	39
SUMMARY.....	40

POPIS SLIKA

Slika 1. Mikrokontroler Microbit (Wikipedia, 2021.).....	11
Slika 2. Mikrokontroler Arduino (Remko van Dokkum – Flickr: Arduino, CC BY 2.0).....	12
Slika 3. Infracrveni detektor prepreka (Diykits.eu 2024.).....	13
Slika 4. Ultrazvučni daljinomjer HC-SR04 (Diykits.eu 2024.).....	14
Slika 5. Shema spajanja senzora HC-SR04 na Microbit.....	21
Slika 6. Shema spajanja senzora HC-SR04 na Arduino.....	23
Slika 7. Shema spajanja H mosta, motora i senzora na Microbit.....	25
Slika 8. Shema spajanja H mosta, motora i senzora na Arduino.....	29
Slika 9. Multiplatformski robot u dijelovima.....	32
Slika 10. Montaža istosmjernih motora na šasiju.....	33
Slika 11. Montaža kugličnog kotača na šasiju.....	33
Slika 12. Shema spajanja svih komponenti multiplatformskog robota.....	34

POPIS TABLICA

Tablica 1. Tablica pinova i spojeva.....	35
--	----

POPIS KODOVA

Kôd 1. Definicije za pokretanje motora na Microbitu.....	17
Kôd 2. Petlja za testiranje motora na Microbitu.....	17
Kôd 3. Definiranje GPIO pinova na Arduinu.....	18
Kôd 4. Postavljanje GPIO pinova na Arduinu.....	18
Kôd 5. Testiranje motora na Arduinu.....	19
Kôd 6. Postavljanje pinova senzora na Microbitu.....	21
Kôd 7. Kreiranje funkcije za mjerenje udaljenosti na Microbitu.....	22
Kôd 8. Testiranje senzora udaljenosti na Microbitu.....	22
Kôd 9. Definiranje pinova senzora udaljenosti na Arduinu.....	23
Kôd 10. Pokretanje serijske komunikacije na Arduinu.....	23
Kôd 11. Testiranje senzora udaljenosti na Arduinu.....	24
Kôd 12. Cjeloviti kod za pokretanje robota sa Microbitom.....	27
Kôd 13. Cjeloviti kod za pokretanje robota sa Arduinom.....	32

1. UVOD

Informatika u osnovnim školama ima ključnu ulogu u pripremi učenika za budućnost u kojoj tehnologija igra sve važniju ulogu. Unatoč tome, obrazovni sustavi u Republici Hrvatskoj i mnogim ostalim državama suočavaju se s izazovima u usklađivanju nastavnih programa s ubrzanim tehnološkim napretkom. Kao nastavnik informatike u osnovnoj školi, kroz rad s učenicima primijetio sam nedostatke u trenutnom načinu podučavanja ovog predmeta. Mnogi učenici pokazuju interes za tehnologiju i programiranje, no zbog zastarjelog programa i ograničenih resursa njihova se znatiželja često ne razvija u dublje razumijevanje. Iako je informatika postala obavezan predmet za 5. i 6. razrede, mnogi programi i dalje osiguravaju samo osnovno razumijevanje tehnologije bez pružanja stvarnih praktičnih iskustava koji bi učenicima omogućili kreativno korištenje tehnologije u različitim područjima. Potreba za reformom u nastavi informatike postaje sve očiglednija jer tradicionalni pristupi ne uspijevaju motivirati učenike te razviti njihove tehničke i kreativne vještine. S obzirom na sve veći značaj STEM (eng. Science, Technology, Engineering and Mathematics) područja, potrebno je istražiti načine kako unaprijediti nastavni plan i program informatike kako bi bio relevantniji za suvremeni svijet.

Cilj ovog rada je prikazati kako bi se za izgradnju suvremenijeg programa informatike mogla uvesti tehnologija poput mikrokontrolera (npr. Microbit i Arduino) koja bi omogućila učenicima da kroz praktične projekte bolje razumiju strukturu i sintaksu različitih programskih jezika. Računala nisu samo alati za izradu prezentacija ili pisanje tekstova, već moćni alati koji mogu biti korišteni za razvoj različitih kreativnih i funkcionalnih rješenja.

Uvođenjem mikrokontrolera i robotike u redovni nastavni proces ili kroz izvannastavnu aktivnost, učenicima bi se pružila prilika da shvate kako tehnologija može biti oblikovana i prilagođena njihovim vlastitim željama i potrebama. Za razliku od tradicionalnih metoda nastave koje često koriste samo osnovne računalne alate, multiplatformski robot temeljen na različitim mikrokontrolerima omogućio bi učenicima da usmjere svoju kreativnost u praktične projekte. Mikrokontroleri kao što su Microbit i Arduino obavljaju istu osnovnu funkciju – pokretanje robota, no svaki od njih koristi različite pristupe u programiranju i upravljanju. Na taj način učenici bi imali priliku upoznati više programskih jezika i razviti vještine kodiranja u različitim okruženjima. Ovaj rad fokusira se na način upotrebe Microbita i Arduina jer navedeni mikrokontroleri omogućuju usklađivanje s kurikulumom Informatike u osnovnim školama pružajući učenicima priliku da kroz različite pristupe savladaju osnovne koncepte programiranja. Navedeni mikrokontroleri omogućili bi učenicima nižih razreda da kroz

blokovsko programiranje razviju osnovno razumijevanje logike kodiranja koje se može proširiti u Python kod – povezujući tako prošlo znanje s novim. Korištenjem Arduina u višim razredima omogućilo bi se uvođenje programskog jezika C, što bi učenicima olakšalo prijelaz u srednjoškolske STEM predmete. Implementacija ovakvih tehnologija omogućila bi učenicima konkretniji uvid u primjenu programiranja te otvorilo prostor za suradnju s drugim predmetima poput matematike i tehničke kulture. Time bi se informatičko obrazovanje povezalo s drugim područjima, a učenicima bi se omogućila šira perspektiva o potencijalu tehnologije. Osnaživanje tehničkih i kreativnih vještina učenika putem praktičnih zadataka ključna je priprema za buduće obrazovanje u STEM područjima, čime se mladima pruža prilika za uspješniji nastavak obrazovanja u srednjoj školi i na fakultetu. Ovaj rad predstavlja priliku za razvoj inovativnog pristupa u nastavi informatike, koji bi pridonio kvalitetnijem obrazovanju i učenicima pružio širu perspektivu o primjeni tehnologije te ih motivirao za daljnje učenje i razvijanje u ovom području.

2. MIKROKONTROLERI

U ovome poglavlju obrađeni su mikrokontroleri, odnosno opisana je njihova primjena i građa, te su navedene osnovne funkcije Microbita i Arduina koji se koriste u školama.

2.1. Općenito o mikrokontrolerima

Mikrokontroleri su mala specijalizirana računala koja se koriste za obavljanje specifičnih zadataka. Koriste se u svim elektroničkim uređajima koji obavljaju neku funkciju – od perilica rublja, hladnjaka do složenih automatiziranih strojeva.

2.2. Osnovna struktura mikrokontrolera

Mikrokontroleri sadrže, poput računala, ključne komponente poput:

- CPU (eng. Central Processing Unit) – kao kod računala CPU na mikrokontrolerima je mozak koji je zadužen za izvršavanje koda pohranjenog u memoriji.
- A/D (eng. Analog-To-Digital) i D/A (eng. Digital-To-Analog) konverteri – Oni pretvaraju analogne signale u digitalne i obrnuto. Služe za rad mikrokontrolera s analognim i digitalnim uređajima.
- memorija – RAM (eng. Random Access Memory) je memorija koja je zadužena za privremeno pohranjivanje podataka tijekom izvršavanja programa. ROM (eng. Read-Only Memory) služi za dugotrajnu pohranu programa. Kod različitih vrsta ROM memorije podaci se mogu mijenjati, a za to je potrebna inačica PROM (eng. Programmable Read-Only Memory) memorije koje se dijele na više vrsta kao što su EPROM (eng. Erasable Programmable Read-Only Memory) i EEPROM (eng. Electrically Erasable Programmable Read-Only Memory). Na određenim mikrokontrolerima može se naći i FLASH memorija u koju se najčešće spremaju programi.
- ulazi i izlazi – Kod mikrokontrolera se najčešće nazivaju GPIO (eng. General Purpose Input Output) i oni služe za komunikaciju mikrokontrolera s vanjskim svijetom. Preko njih mikrokontroleri mogu očitavati razne senzore ili slati različite naredbe drugim uređajima te tako upravljati njima.
- komunikacijski moduli - Oni omogućuju komunikaciju s drugim uređajima putem raznih protokola kao što su UART (eng. Universal Asynchronous Receiver-Transmitter), I2C (eng. Inter-Integrated Circuit), SPI (eng. Serial Peripheral Interface)

2.3. Microbit

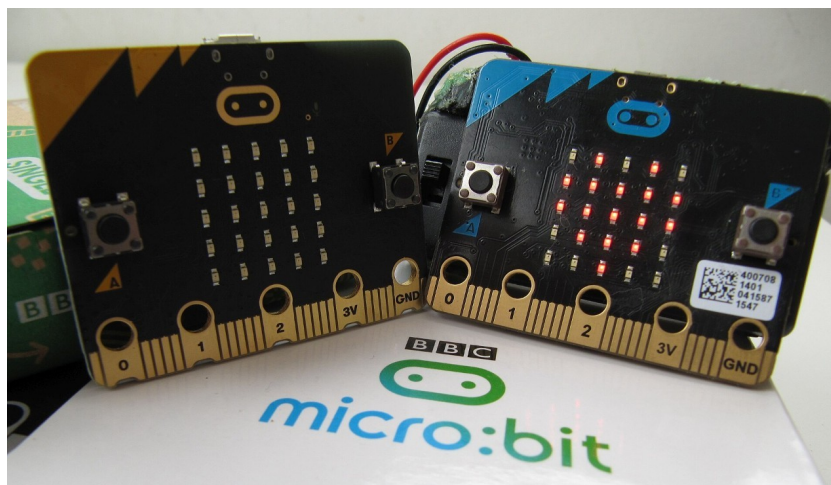
Microbit (Slika 1) je popularni mikrokontroler koji je najrasprostranjeniji u osnovnim školama te se često koristi u izvannastavnim aktivnostima kao što je robotika. Razvijen je od

strane BBC-a (eng. British Broadcasting Corporation). Malih je dimenzija, a odlikuje ga dosta velik broj GPIO pinova (ulazi i izlazi) uz pomoć kojih komunicira s drugim uređajima. Na prednjoj strani ima LED (eng. Light Emitting Diode) matricu dimenzija 5x5 uz pomoć kojih može ispisati osnovna očitavanja, nacrtati neke osnovne oblike ili prikazati neke animacije. Također, na prednjoj strani nalaze se dva gumba koji su potpuno programabilni te se najčešće mogu koristiti kao okidači za određeni dio koda. Označeni su slovima A i B te podržavaju pritisak u isto vrijeme, što omogućuje tri različite kombinacije. Od senzora ima ugrađene akcelerometar i kompas koji omogućuju registraciju nagiba i orijentacije mikrokontrolera. Pomoću njih mogu se napraviti okidači u programu kao npr. da pod određenim kutom ili prilikom protresanja odradi dio koda. Od bežične komunikacije ima Bluetooth i radio komunikaciju te preko nje može ostvariti vezu između dvaju ili više Microbita i računala. Isto tako, sadrži 25 GPIO pinova uz pomoć kojih može upravljati drugim uređajima ili dobivati očitavanja s različitih senzora. Oni se nalaze na rubovima štampane pločice mikrokontrolera i izgledaju kao računalna komponenta. Za njihovo korištenje najčešće se treba uzeti vanjska pločica u koju se Microbit umetne i preko nje se pretvori u pinove na koje se spajaju drugi uređaji ili senzori. Neki od tih pinova podržavaju PWM (eng. Pulse Width Modulation) izlaz, I2C, SPI i druge funkcije te napajanje i od 3,3 V i GND (eng. Ground). Microbit se može napajati putem USB (eng. Universal Serial Bus) priključka, vanjskih baterija i određenih GPIO pinova, što ga čini savršenim za korištenje u nastavi Informatike u osnovnim školama. Microbit se može programirati na više načina koristeći različita razvojna okruženja kao što su:

- MicroPython – inačica Pythona prilagođena za programiranje u Python programskom jeziku. Idealna je za otkrivanje različitih apstraktnih mogućnosti Pythona te samim time učenici mogu razviti širu sliku programiranja.
- JavaScript – Koriste ga napredniji korisnici.
- C/C++ – Koriste ga napredniji programeri i samim time Microbit se može koristiti za profesionalne potrebe.
- Block Editor (Microsoft MakeCode) – Vizualno programiranje koje koristi blokove slične programskom jeziku Scratch. Ovo je idealno okruženje za učenike nižih razreda. Uređaj se programira stavljanjem različitih blokova naredbi koji predstavljaju različite funkcije. također, praktičan je za prelazak na Python ili Javascript budući da se u njemu kod može pogledati u navedenim sintaksama.

Uz pomoć Microbita sama nastava Informatike u osnovnim školama može postati zanimljiva budući da preko njega učenici mogu vidjeti rezultate svog programiranja. Spajanjem različitih

senzora ili uređaja na Microbit učenici mogu razvijati svoje ideje te samim time i svoju inovativnost.



Slika 1. Mikrokontroler Microbit (Wikipedia, 2021.)

2.4. Arduino

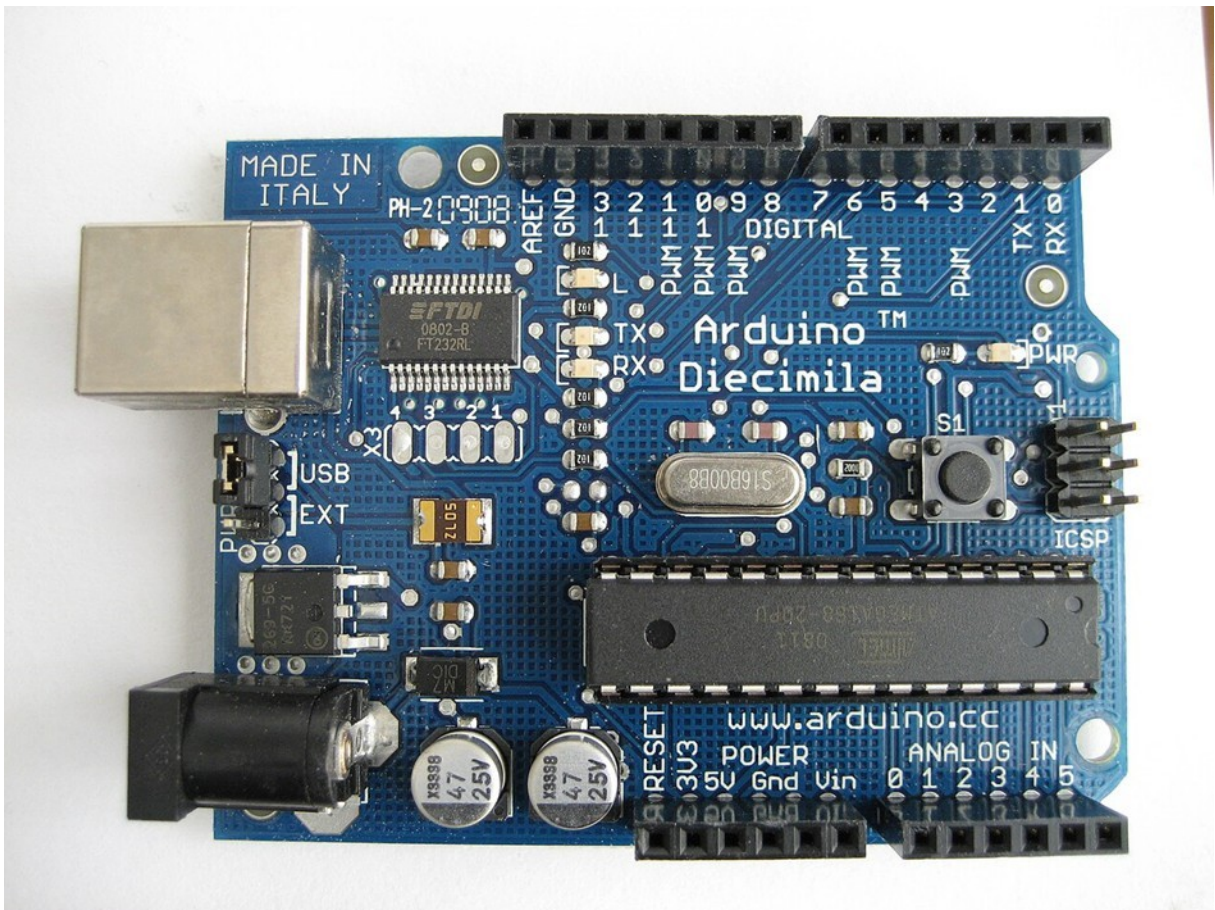
Arduino (Slika 2) je *open source* hardverska i softverska platforma. Zbog velikog broja različitih vrsta Arduino ploča, često se koristi za razvoj sustava. U osnovnim školama ne koristi se toliko često jer se programiraju u C/C++ varijantama programskog jezika koji nije u planu i programu nastave Informatike. Razvojno okruženje Arduina je Arduino IDE (eng. Integrated Development Environment). To je besplatni program preko kojeg se programi prebacuju na Arduino i mogu se pratiti stanja ulaza i izlaza Arduino mikrokontrolera. Kao i svi mikrokontroleri, Arduino ima skup analognih i digitalnih pinova koji se koriste kao ulazi i izlazi. Putem njih se Arduino povezuje na različite vanjske uređaje ili senzore. Arduino pločice na sebi imaju ugrađen mikrokontroler koji su najčešće Atmel serije. Arduino pločice mogu sadržavati i druge mikrokontrolere kao što su: ARM (eng. Acorn RISC Machine) Cortex-M, ESP32 (eng. Espressif Systems), ESP8266. Isto tako, postoji više varijanti Arduino pločica od kojih razlikujemo:

- Arduino Uno – Ona je najpopularnija i najčešće korištena ploča. Idealna je za početnike. Bazira se AT (eng. Attention) mega 328P kontroleru i ima 14 digitalnih pinova i 6 analognih ulaza. Preko njih se spaja s drugim sensorima i uređajima.
- Arduino Nano – Mala i kompaktna verzija Arduina koja ima iste funkcije kao Arduino Uno. Često se zbog njenih dimenzija koristi za ugradbene projekte.
- Arduino Mega – Pločica koja sadrži više pinova i ima veći kapacitet memorije, bazirana je na ATmega2560 mikrokontroleru. Budući da ima više pinova, idealna je

za korištenje kod projekata koji zahtijevaju veći broj senzora ili upravljaju većim brojem uređaja.

- Arduino Leonardo – Varijanta Arduino ploče koja može emulirati USB uređaje kao što su tipkovnica ili miš i bazirana je na ATmega32u4 mikrokontroleru.

Samim time što ima više varijanti Arduino ploča, primjena je dosta široka. Arduino se dosta često koristi u srednjim školama jer spaja naprednu elektroniku i programiranje. Zbog svoje fleksibilnosti, često se može naći u prototipovima elektroničkih uređaja. Kako je Arduino *open source*, jako puno ljudi koristi ga za hobi projekte i ima široku podršku zajednice. U nastavi Informatike može se koristiti kao alat za osnovno učenje C i C++ jezika. Za Arduino platformu postoji mnoštvo dodataka koji ga čine moćnom platformom za razvijanje novih i inovativnih projekata.



Slika 2. Mikrokontroler Arduino (Remko van Dokkum – Flickr: Arduino, CC BY 2.0)

3. KOMPONENTE MULTIPLATFORMSKOG ROBOTA

Robotika je u sklopu nastave Informatike često ograničena sastavljenim robotima u koje se umetne mikrokontroler kako bi se dalje programiralo. Ideja multiplatformskog robota je da se nauče i sve ostale komponente koje ga pokreću i na koji način radi.

3.1. Glavni dijelovi robota

Senzori robota ključni su elementi dizajna robota. Oni služe za očitavanje okoline u kojoj se robot nalazi. Roboti koji su kreirani za samostalno kretanje svoju putanju određuju po određenim informacijama koje dobivaju sa senzora. Jedan od glavnih senzora koji se koriste u pokretnom robotu su senzori za prepreke. Motori su ključne komponente robota koje mu omogućuju kretanje. Oni mogu biti jednostavni u slučaju da obavljaju osnovne funkcije, i komplicirani, za veće i složenije manipulacije i manevre. Za jednostavno pokretanje robota najčešće se koriste istosmjerni motori, dok se za preciznije kretanje koriste servo motori. Za njihovo pokretanje, budući da su motori elektrotehničke komponente, koriste se razni kontroleri motora ovisno o motoru i primjeni. Najčešći način pokretanja istosmjernih motora je pomoću H-mosta. To je sklop koji se koristi za kontrolu smjera rotacije i brzine kod istosmjernih motora. Ime je dobio po tome što, kada se nacrtava u shemi, nalikuje na slovo H. On omogućuje da struja prolazi u oba smjera kroz motor, što omogućuje okretanje motora u jednu i drugu stranu zamjenom polariteta na spojevima elektromotora.

3.1.1. Senzori prepreka

Infracrveni senzori prepreka (Slika 3) koriste infracrvenu svjetlost koju infracrvena dioda emitira i reflektira se od raznih površina. Uz pomoć infracrvenog primatelja određuje se udaljenost od određenog objekta mjerenjem reflektirane svjetlosti. Prednosti kod takve vrste senzora su brza reakcija budući da brzina svjetlosti iznosi otprilike $3 * 10^8$ metara u sekundi. Samim time, infracrveni senzori brže mogu očitati prepreku koja se nalazi ispred njih. Infracrveni senzori isto tako imaju dobro očitavanje različitih vrsta materijala. Njihove mane su to što uvjeti osvjetljenja mogu značajno utjecati na očitavanje. Također, površine koje su crne ili mat slabo reflektiraju svjetlost, pa samim time i otežavaju detekciju predmeta.



Slika 3. Infracrveni detektor prepreka (Diykits.eu 2024.)

Ultrazvučni senzori (Slika 4) koriste ultrazvučne valove kako bi očitali prepreke. Sastoje se od odašiljača i prijammika. Odašiljač se sastoji od piezoelektričnog materijala koji pretvara električnu energiju u zvučni val visoke frekvencije. Prijamnik je komponenta koja prima reflektirane zvučne valove i pretvara zvuk u električnu energiju. Budući da koriste zvuk za mjerenje udaljenosti, na njih ne utječe pozadinsko osvjetljenje. Imaju veći opseg detekcije od infracrvenih senzora pa mogu očitati prepreku i na većim udaljenostima. Problem ultrazvučnog senzora je što teže očitava male i porozne površine, zbog čega može doći do smanjenja točnost mjerenja.



Slika 4. Ultrazvučni daljinomjer HC-SR04 (Diykits.eu 2024.)

4. PRAKTIČNI RAD – MULTIPLATFORMSKI ROBOT ZA UNAPREĐENJE NASTAVE INFORMATIKE U OSNOVNIM ŠKOLAMA

U ovome poglavlju detaljno se opisuje sastavljanje robota, programiranje senzora i H mosta te analiza razlike kodova sa istim funkcijama na različitim platformama.

4.1. L298N Dual H Most

Za pokretanje motora robota koristi se L298N H-most. On se sastoji od integriranog kruga koji služi za kontrolu smjera i brzine dva motora. Maksimalna struja po motoru je 2 A, što ga čini dovoljnim za ovaj projekt. Na pločici postoje priključci za dva motora. Lijevi motor spaja se na priključke OUT (eng. Output) 1 i OUT 2, a desni na OUT 3 i OUT 4. Za spajanje H mosta na mikrokontroler koriste se IN (eng. Input) 1 i IN 2 koji se spajaju na GPIO pinove mikrokontrolera za kontrolu smjera rotacije lijevog motora, dok IN 3 i IN 4 na GPIO pinove mikrokontrolera za kontrolu smjera rotacije desnog motora. ENA i ENB se spajaju na PWM signal mikrokontrolera za regulaciju brzine rotacija motora. Ako se ne želi kontrolirati brzina rotacije, ENA i ENB spaja se na 5 V. Time se dobiva puna brzina rotacije motora. VCC (eng. Voltage at the Common Collector) i GND priključci su za napon motora, ovisno o tome koji je radni napon motora. Smjer motora kontrolira se na sljedeći način: ako je na ulaz IN1 pušten signal, a na ulaz IN2 nije, motor se vrti u jednom smjeru. Kada se signali na ulazima zamijene tako da na IN1 ne pušta signal dok na IN2 pušta, motor se vrti u drugom smjeru. Motori su zaustavljeni kada se ni na IN1 ni na IN2 ne pusti signal. Ako se na oba ulaza pusti signal, motor se zaustavlja, ali na taj način realizira se kočenje. Isti način koristi se za drugi motor, samo što se signali puštaju na ulaze IN3 i IN4. Sa vrijednostima od 0-255 na ulazima ENA i ENB kontrolira se brzina vrtnje motora. Ako je vrijednost 0, motori su zaustavljeni. Ako je vrijednost 255, motori se vrte najvećom brzinom.

4.1.1. Upravljanje motorima sa Microbitom

Kako bi se motori mogli pokretati s Microbitom, potrebno je koristiti Microbit extender pločicu te sve spojiti na sljedeći način:

- IN1 s modula na PIN 0 Microbita.
- IN2 s modula na PIN 1 Microbita.
- IN3 s modula na PIN 2 Microbita.
- IN4 s modula na PIN 8 Microbita.
- ENA s modula na PIN 12 Microbita (PWM za motor A).
- ENB s modula na PIN 13 Microbita (PWM za motor B).

- GND s modula na GND pin Microbita.

Kako bi se mogao koristiti modul Microbit koji sadrži sve funkcije i klase potrebne za programiranje microbita u MicroPythonu, potrebno ih je uvesti na početku koda s naredbom *from microbit import **. Implementacijom ovog modula daje se mogućnost kontrole pinova s naredbom *write_digital*. U početnoj fazi upravljanja motorima potrebno je staviti i precizna vremenska ograničenja kako bi se motori mogli kontrolirati određeno vrijeme, te se treba uvesti i modul *time* s naredbom *import time*. Nakon implementacije tih modula potrebno je definirati GPIO pinove Microbita koje koristimo kao kontrolu H mosta. Linijom koda *IN1=pin0* definira se da je pin 0 s Microbita spojen na IN1 pin H mosta. Ostali se definiraju na isti način. *IN2=pin1*, *IN3=pin2*, *IN4=pin8*. IN4 je spojen na pin 8 microbita budući da su se na ostalim pinovima stvarale neke smetnje. Za kontrolu brzine koriste se pinovi 12 i 13 te se definiraju kodom *ENA=pin12* i *ENB=pin13*. Idući korak je definicija funkcija za pokretanje motora koja se kasnije u kodu poziva. Funkcije se definiraju kako je opisano u Kôd 1.

```
def motorA_forward():
    ENA.write_digital(1) # Omogući motor A
    IN1.write_digital(1) # Postavi smjer naprijed
    IN2.write_digital(0)

def motorA_backward():
    ENA.write_digital(1) # Omogući motor A
    IN1.write_digital(0) # Postavi smjer natrag
    IN2.write_digital(1)

def motorA_stop():
    ENA.write_digital(0) # Zaustavi motor A

def motorB_forward():
    ENB.write_digital(1) # Omogući motor B
    IN3.write_digital(1) # Postavi smjer naprijed
    IN4.write_digital(0)
```

```

def motorB_backward():
    ENB.write_digital(1) # Omogući motor B
    IN3.write_digital(0) # Postavi smjer natrag
    IN4.write_digital(1)

def motorB_stop():
    ENB.write_digital(0) # Zaustavi motor B

```

Kôd 1. Definicije za pokretanje motora na Microbitu

Za testiranje motora može se izraditi jednostavna petlja koja prvo okreće motore u jednom smjeru 2 sekunde, stane na sekundu, promjeni smjer rotacije motora na 2 sekunde i onda ponovo zaustavi motore. Budući da se kretanje motora mora testirati kontinuirano, stavlja se u petlju. Kod za testiranje motora opisan je u Kôd 2.

```

while True:
    motorA_forward()
    motorB_forward()
    sleep(2000) # Vozi naprijed 2 sekunde

    motorA_stop()
    motorB_stop()
    sleep(1000) # Stani na 1 sekundu

    motorA_backward()
    motorB_backward()
    sleep(2000) # Vozi unatrag 2 sekunde

    motorA_stop()
    motorB_stop()
    sleep(1000) # Stani na 1 sekundu

```

Kôd 2. Petlja za testiranje motora na Microbitu

Testiranje motora prekida se tako da se Microbit ugasi, budući da se navedena petlja neprestano izvodi.

U ovome kodu nije napravljeno testiranje brzine. Ono se lagano može napraviti sljedećim

promjenama:

U ENA i ENB se umjesto `write_digital()` stavlja `write_analog()` zato što microbit koristi 10-bitni PWM stoga je maksimalna vrijednost koja se koristi 1023. Primjer koda koji bi promijenio brzinu motora na približno 50% je `write_analog(512)`.

4.1.2. Upravljanje motorima s Arduinom

Budući da je Arduino različit od Microbita, motor se spaja na njega po sljedećoj shemi:

- IN1 s modula na PIN 8 Arduina
- IN2 s modula na PIN 9 Arduina
- IN3 s modula na PIN 10 Arduina
- IN4 s modula na PIN 11 Arduina
- ENA s modula na PIN 5 Arduina (PWM za motor A)
- ENB s modula na PIN 6 Arduina (PWM za motor B).

Prvo se trebaju definirati GPIO pinovi koji se koriste na Arduinu na način opisanim u Kôd 3.

```
int IN1 = 9;
int IN2 = 8;
int ENA = 10;
int IN3 = 7;
int IN4 = 6;
int ENB = 5;
```

Kôd 3. Definiranje GPIO pinova na Arduinu

Tu je postavljeno koji se pinovi koriste u programu. Idući korak je postavljanje GPIO pinova kao izlazni. To se izvodi kodom opisanim u Kôd 4.

```
void setup() {
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(ENA, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(ENB, OUTPUT);
}
```

Kôd 4. Postavljanje GPIO pinova na Arduinu

Nakon postavljanja pinova i načina na koji će pinovi funkcionirati, radi se petlja koja testira kretanje motora. Testiranje će se odvijati na isti način kao i kod Microbita: prvo se u

neprekidnoj petlji motori pokreću u jednom smjeru dvije sekunde, zaustavljaju na jednu sekundu, okreće se smjer na dvije sekunde i zaustavljaju na jednu sekundu. Kod te petlje opisan je u Kôd 5.

```
void loop() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 255);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 255);
    delay(2000);
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(1000);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 255);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENB, 255);
    delay(2000);
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(1000);
}
```

Kôd 5. Testiranje motora na Arduinu

Budući da je ovaj kod stavljen u neprekidnu petlju *void loop()*, testiranje motora se prekida gašenjem Arduina.

4.1.3. Analiza razlike koda pokretanja motora za Microbit i Arduino

Oba koda rade identičnu stvar, no sintaksa je potpuno drugačija. Time se može napraviti ishod u nastavi Informatike u kojem učenici mogu zaključiti sljedeće:

- Microbit koristi MicroPython izvedbu programskog jezika Python, dok Arduino koristi C++ programski jezik.

- Za upravljanje GPIO izlazima Microbita u Pythonu se koristi naredba `write_digital()` ili `write_analog()`, dok je kod Arduina `digitalWrite()` i `analogWrite()` sintaksa drugačija, ali ostvaruje isti rezultat.
- Na Microbitu se GPIO pinovi ne trebaju definirati kao ulazni ili izlazni, dok se na Arduinu moraju.
- Microbit ima vrijednosti na `write_analog()` od 0 do 1023 za razliku od Arduina koji ima `analogWrite()` od 0-255, što znači da Microbit ima veću preciznost.
- Python je jednostavniji i lakši za čitanje i pisanje od C++.

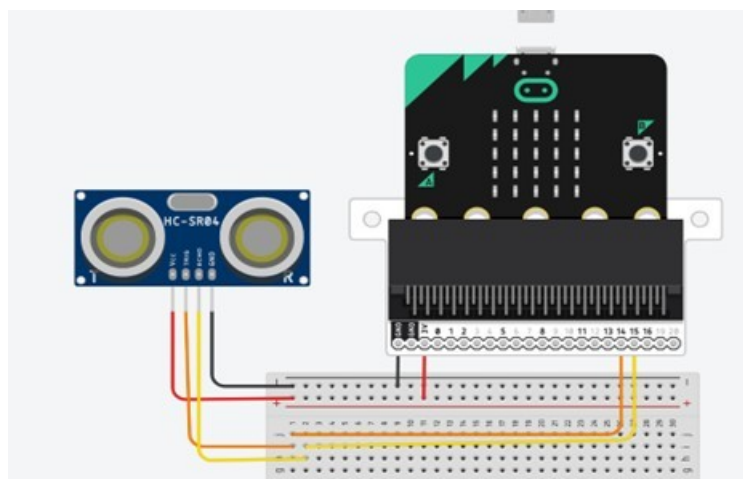
4.2. Ultrazvučni daljinomjer HC-SR04

Kao osnovni senzor za mjerenje udaljenosti od objekata i prepreka, na ovom projektu najbolje je koristiti ultrazvučni HC-SR04 senzor. Njegova cijena, jednostavnost i dostupnost idealna je za primjenu u osnovnim školama. Budući da koristi zvuk, može se iskoristiti i za dokazivanje eksperimenata u nastavi Fizike. Senzor ima četiri pina. Na VCC pin se spaja napajanje (3.3 V ili 5 V). Na GND se spaja masa. TRIG pin služi za slanje ultrazvučnih valova, dok ECHO služi za primanje reflektiranih ultrazvučnih valova. Funkcionira na način da se na TRIG pin pošalje kratki signal koji signalizira modulu da pošalje ultrazvučni val. Kada se ultrazvučni val vrati na modul refleksijom od površine, modul na pin ECHO šalje vrijeme koje je bilo potrebno za val pri čemu se onda računa udaljenost. Formula za računanje udaljenosti je $\frac{\text{vrijeme} * \text{brzinazvuka}}{2}$. U ovom slučaju mora se podijeliti s 2 zato što zvuk prelazi duplu udaljenost – od izvora do površine i onda natrag do modula.

4.2.1. Korištenje senzora na Microbitu

Kako bi se senzor mogao koristiti Microbitom, potrebno ga je spojiti kako je prikazano na Slika 5:

- VCC pin modula spojiti na 3V pin Microbita
- GND pin modula spojiti na GND pin Microbita
- TRIG pin modula spojiti na PIN14 GPIO Microbita
- ECHO pin modula spojiti na PIN15 GPIO Microbita



Slika 5. Shema spajanja senzora HC-SR04 na Microbit

Nakon što je senzor spojen na Microbit, potrebno je napisati kod koji će dobivati očitavanja senzora i pretvoriti ih u udaljenost. Prvi korak je implementacija *Microbit library-a* da bi se moglo koristiti Microbit GPIO pinove. To se radi pomoću naredbe *from microbit import **. Budući da se mora mjeriti i vrijeme, potrebno je implementirati *time* s linijom koda *import time*. Nakon toga definiraju se pinovi na koje je senzor spojen sa microbitom na način opisan u Kôd 6.

```
TRIG = pin14  
ECHO = pin15
```

Kôd 6. Postavljanje pinova senzora na Microbitu

Kako bi se poslao zvuk, potrebno je aktivirati izlaz na pinu 14. Prvo se pin postavlja u stanje 0, čeka dvije mikrosekunde, nakon toga se postavlja pin u stanje 1 na deset mikrosekundi i nakon toga ga se vraća u stanje 0. Nakon poslanog signala, na pinu 15 očitava se aktivnost koja se stvara refleksijom zvučnog signala od površine. Mjeri se koje je vrijeme potrebno da se signal detektira i to se zapisuje u varijablu *start_time*. Dokle je signal na pinu 15 aktivan, vrijeme se zapisuje u varijablu *end_time*. Nakon toga je potrebno izračunati koliko je vremena prošlo od promjena. Uzima se u obzir da je udaljenost kada se signal šalje dupla, budući da putuje od izvora do površine i natrag do senzora. To sve se definira u funkciji *measure_distance()* opisanoj u Kôd 7 koja na kraju vraća udaljenost.

```
def measure_distance():  
    # Pošalji TRIG signal  
    TRIG.write_digital(0)  
    time.sleep_us(2)  
    TRIG.write_digital(1)
```

```

time.sleep_us(10)
TRIG.write_digital(0)
# Očitaj ECHO signal
while ECHO.read_digital() == 0:
    start_time = time.ticks_us()
while ECHO.read_digital() == 1:
    end_time = time.ticks_us()
# Izračunaj udaljenost u cm
duration = time.ticks_diff(end_time, start_time)
distance = (duration * 0.0343) / 2
return distance

```

Kôd 7. Kreiranje funkcije za mjerenje udaljenosti na Microbitu

Za testiranje senzora i preciznosti koristi se 5x5 led matrica Microbita na kojoj se ispisuje udaljenost. Budući da se kod treba cijelo vrijeme odvijati, koristi se petlja *while True* opisana u Kôd 8.

while True:

```

# Mjerenje udaljenosti
distance = measure_distance()
# Prikaz na LED ekranu micro:bit-a
display.scroll(str(distance) + " cm")
sleep(500)
# Pauza između mjerenja

```

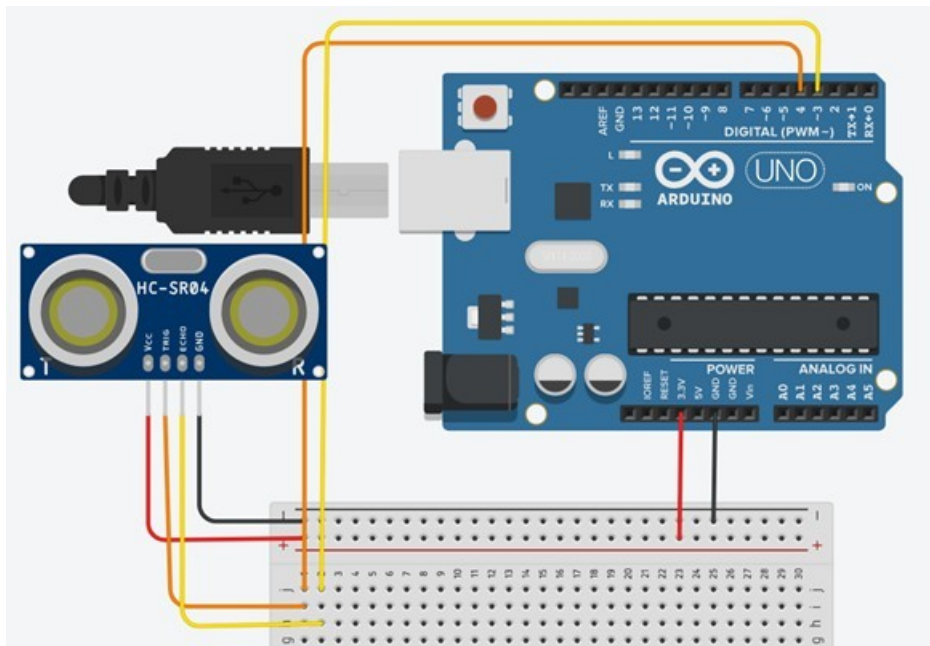
Kôd 8. Testiranje senzora udaljenosti na Microbitu

Kroz pokretanje ovog koda verificira se radi li senzor dobro.

4.2.2. Korištenje senzora na Arduinu

Kako bi se senzor mogao koristiti na Arduinu, potrebno ga je spojiti kako je prikazano na Slika 6:

- VCC pin s modula na 3.3V pin Arduina
- GND pin s modula na GND pin Arduina
- TRIG pin sa modula na pin 4 Arduina
- ECHO pin sa modula na pin 3 Arduina



Slika 6. Shema spajanja senzora HC-SR04 na Arduino

Nakon što je senzor spojen na Arduino, u kodu nije potrebno implementirati dodatne biblioteke jer senzor koristi standardne funkcije arduina kao što su *pinMode()*, *digitalWrite()*, *pulseIn()* i *Serial*, a one su uključene u osnovni Arduino paket. Potrebno je definirati koji će se pinovi koristiti kodom opisanim u Kôd 9.

```
const int trigPin = 4;
const int echoPin = 3;
```

Kôd 9. Definiranje pinova senzora udaljenosti na Arduino

Nakon toga potrebno je odrediti hoće li GPIO pinovi koji se koriste biti ulazni ili izlazni. Budući da Arduino mora poslati signal senzoru na TRIG pin preko pina 4, njega je potrebno definirati kao izlaznog. Echo pin prima signal, što znači mora biti ulazni. Kako bi se pratili rezultati putem serijskog monitora, potrebno je pokrenuti serijsku komunikaciju. To se definira kodom opisanim u Kôd 10.

```
void setup() {
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
}
```

Kôd 10. Pokretanje serijske komunikacije na Arduino

Za mjerenje udaljenosti koristi se ista logika programa kao i za Microbit, odnosno potrebno je poslati signal, izmjeriti vrijeme potrebno da se vrati refleksija i uz pomoć brzine zvuka dobiti ukupnu udaljenost koju je zvučni val prošao. Potrebno je tu udaljenost podijeliti s brojem 2 jer putuje od senzora do prepreke i od prepreke natrag do senzora. Cijeli kod mora se neprestano odvijati kako bi se senzor testirao, te se zatim stavlja u loop. Kod koji to izvodi je opisan u Kôd 11.

```
void loop() {  
    long duration; int distance; // Slanje ultrazvučnog impulsa  
    digitalWrite(trigPin, LOW); // Postavlja pin trig na 0  
    delayMicroseconds(2); // Kratka pauza  
    digitalWrite(trigPin, HIGH); // Postavlja pin trig na 1  
    delayMicroseconds(10); // Slanje impulsa 10 mikrosekundi  
    digitalWrite(trigPin, LOW); // Očitavanje trajanja echo signala  
    duration = pulseIn(echoPin, HIGH); //Mjeri vrijeme koliko je  
    //pin ECHO 1  
    distance = (duration * 0.0343) / 2; //Izračun udaljenosti i  
    //zapisivanje u distance  
    Serial.print("Udaljenost: ");  
    Serial.print(distance);  
    Serial.println(" cm");  
    delay(500); // Ispisuje udaljenost svakih pola sekunde  
}
```

Kôd 11. Testiranje senzora udaljenosti na Arduinu

4.3. Izrada platforme robota za više mikrokontrolera

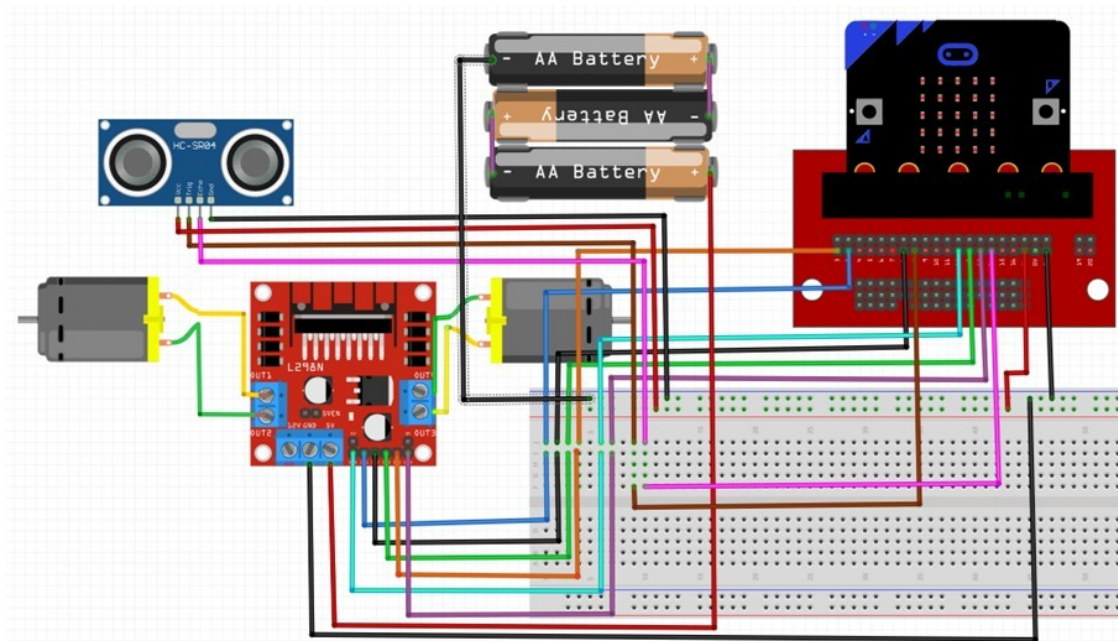
Za pokretanje robota s microbitom koristi se shema spajanja L298N H Mosta kako je prikazano na Slika 7.

- IN1 – PIN 0 na Microbitu (kontrola motora lijevo)
- IN2 – PIN 1 na Microbitu (kontrola motora lijevo)
- IN3 – PIN 2 na Microbitu (kontrola motora desno)
- IN4 - PIN 3 na Microbitu (kontrola motora desno)
- ENA – povezuje se na 5 V
- ENB – povezuje se na 5 V
- Motor A – spojeni s motorom lijevo

- Motor B – spojeni s motorom desno

HC-SR04 Ultrazvučni senzor:

- TRIG – PIN 8 na Microbitu
- ECHO – PIN 14 na Microbitu
- VCC – 5 V (ili 3.3 V ako se koristi napajanje s microbitom)
- GND – GND na Microbitu



Slika 7. Shema spajanja H mosta, motora i senzora na Microbit

Nakon spajanja Microbita kako je prikazano na Slika 7, sa programom opisanim u Kôd 12 se upravlja pokretanjem motora uz pomoć H mosta i ultrazvučnog senzora. Kod uz pomoć kojeg se upravlja microbit napisan je u MicroPython programskom jeziku.

```
from microbit import *
import utime

# Pinovi za L298N
IN1 = pin0
IN2 = pin1
IN3 = pin2
```

```

IN4 = pin3

# Pinovi za HC-SR04
TRIG = pin8
ECHO = pin14

# Funkcija za mjerenje udaljenosti
def get_distance():
    TRIG.write_digital(0)
    utime.sleep_us(2)
    TRIG.write_digital(1)
    utime.sleep_us(10)
    TRIG.write_digital(0)

    pulse_start = utime.ticks_us()
    while ECHO.read_digital() == 0:
        pulse_start = utime.ticks_us()

    pulse_end = utime.ticks_us()
    while ECHO.read_digital() == 1:
        pulse_end = utime.ticks_us()

    pulse_duration = pulse_end - pulse_start
    distance = (pulse_duration * 0.0343) / 2 # Udaljenost u cm
    return distance

# Funkcija za vođenje motora
def drive_forward():
    IN1.write_digital(1)
    IN2.write_digital(0)
    IN3.write_digital(1)
    IN4.write_digital(0)

```

```

def stop_motors():
    IN1.write_digital(0)
    IN2.write_digital(0)
    IN3.write_digital(0)
    IN4.write_digital(0)

def reverse():
    IN1.write_digital(0)
    IN2.write_digital(1)
    IN3.write_digital(0)
    IN4.write_digital(1)
    utime.sleep(2) # Povuci se 2 sekunde

# Glavni program
while True:
    distance = get_distance()
    display.show(Image.SQUARE)

    if distance > 10: # Ako je udaljenost veća od 10 cm
        drive_forward()
    else:
        stop_motors()
        utime.sleep(3) # Čekaj 3 sekunde
        reverse()
        stop_motors()
        utime.sleep(1) # Kratki odmor prije ponovnog kretanja

```

Kôd 12. Cjeloviti kod za pokretanje robota sa Microbitom

Objašnjenje koda:

- *get_distance* funkcija: Ova funkcija koristi HC-SR04 za mjerenje udaljenosti od prepreke
- *drive_forward* funkcija: Pomiče robota naprijed

- *stop_motors* funkcija: Zaustavlja robota
- *reverse* funkcija: Pomiče robota unatrag
- glavni dio programa: Provjerava udaljenost. Ako je prepreka bliža od 10 cm, robot se zaustavlja i nakon 3 sekunde se povlači unatrag.

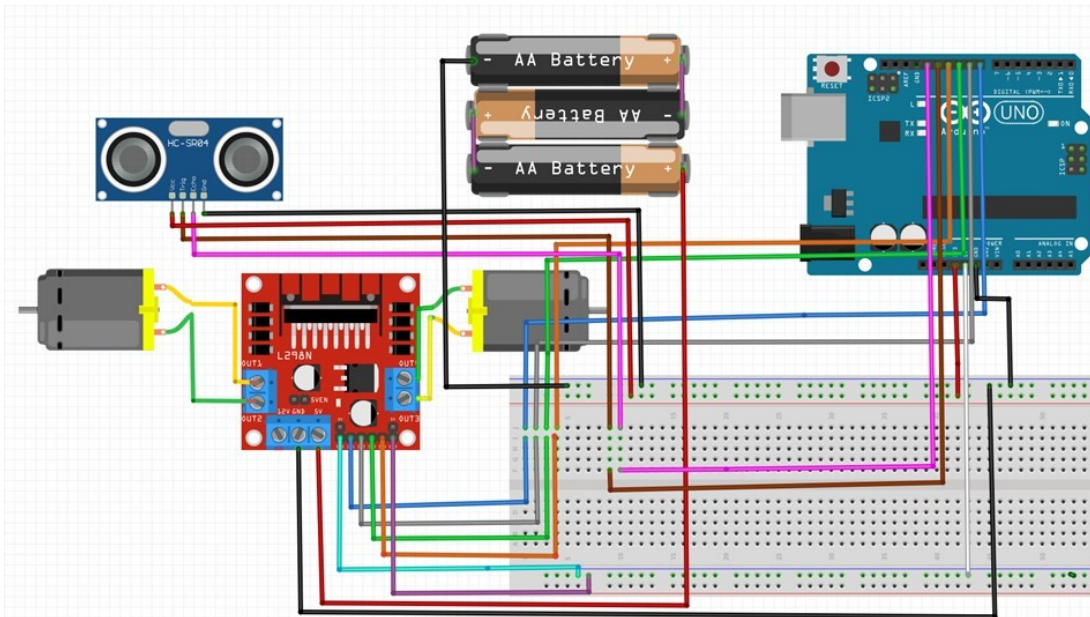
4.4. Pokretanje s Arduinom

Za pokretanje robota s Arduino Uno koristi se shema spajanja L298N H mosta kako je prikazano na Slika 8.

- IN1 – PIN 8 na Arduinu
- IN2 – PIN 9 na Arduinu
- IN3 – PIN 10 na Arduinu
- IN4 – PIN 11 na Arduinu
- ENA – Povezati s 5V pinom na Arduinu (ili s PWM pinom ako se želi regulirati brzina)
- ENB – Povezati s 5V pinom na Arduinu (ili s PWM pinom ako se želi regulirati brzina)
- Motor A – spojeni s motorom lijevo
- Motor B – spojeni s motorom desno

HC-SR04 Ultrazvučni senzor:

- TRIG – PIN 12 na Arduinu
- ECHO – PIN 13 na Arduinu
- VCC – 5V pin na Arduinu
- GND – GND na Arduinu



Slika 8. Shema spajanja H mosta, motora i senzora na Arduino

Nakon spajanja Arduina na ovaj način, sa kodom opisanim u Kôd 13 se upravlja pokretanje motora uz pomoć H mosta i ultrazvučnog senzora. Kod uz pomoć kojeg se upravlja Arduinoom napisan je u C++ za Arduino programskom jeziku.

```
// Pinovi za L298N
```

```
const int IN1 = 8;
```

```
const int IN2 = 9;
```

```
const int IN3 = 10;
```

```
const int IN4 = 11;
```

```
// Pinovi za HC-SR04
```

```
const int TRIG = 12;
```

```
const int ECHO = 13;
```

```
// Funkcija za mjerenje udaljenosti
```

```
long getDistance() {
```

```

// Izmjeri udaljenost pomoću HC-SR04
digitalWrite(TRIG, LOW);
delayMicroseconds(2);
digitalWrite(TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG, LOW);

long duration = pulseIn(ECHO, HIGH);
long distance = (duration * 0.0343) / 2; // Udaljenost u cm
return distance;
}

// Funkcija za vođenje motora
void driveForward() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void stopMotors() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}

```

```

}

void reverse() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(2000); // Povuci se 2 sekunde
}

void setup() {
    // Postavi pinove kao ulaze/izlaze
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(TRIG, OUTPUT);
    pinMode(ECHO, INPUT);

    Serial.begin(9600);
}

void loop() {
    long distance = getDistance();
    Serial.print("Distance: ");

```



```

Serial.println(distance);

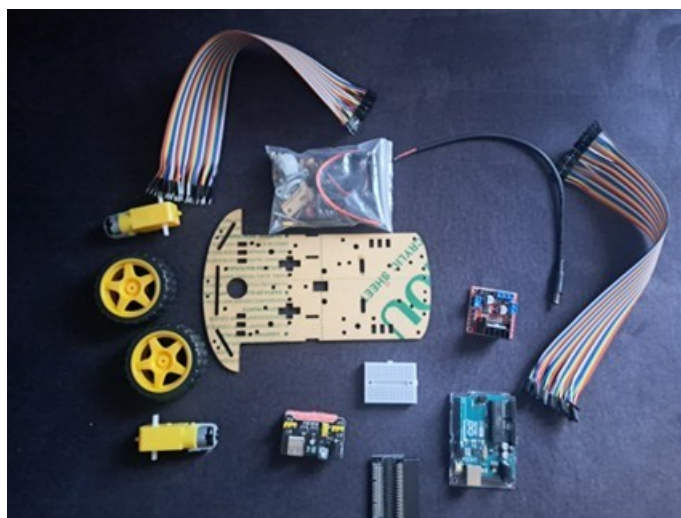
if (distance > 10) { // Ako je udaljenost veća od 10 cm
    driveForward();
} else {
    stopMotors();
    delay(3000); // Čekaj 3 sekunde
    reverse();
    stopMotors();
    delay(1000); // Kratki odmor prije ponovnog kretanja
}
}

```

Kôd 13. Cjeloviti kod za pokretanje robota sa Arduinoom

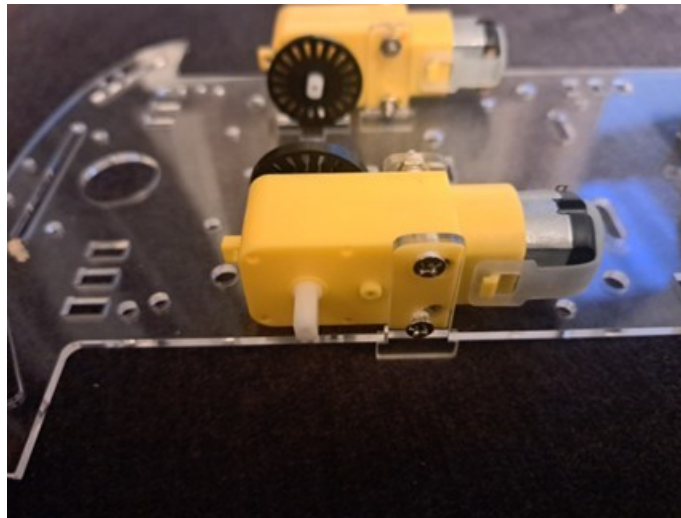
4.5. Sastavljanje završnog multiplatformskog robota

Budući da je multiplatformski robot namijenjen korištenju u osnovnim školama, za robotsko vozilo je korišteno široko dostupni Smart Robot Car paket za Arduino. To je vozilo na kojem se svaki element mora montirati. Na Slika 9 se vide sve komponente multiplatformskog robota.



Slika 9. Multiplatformski robot u dijelovima

Prvo se nosači motora postavljaju kroz za to predviđenu rupu na glavnoj šasiji vozila zatim se montiraju istosmjerni motori koji ne dopuštaju dodatno gibanje nosača te ih uglavljaju u poziciju kako je prikazano na Slika 10. Nakon montaže motora, na njihove osovine se postavljaju gumeni kotači.



Slika 10. Montaža istosmjernih motora na šasiju

Za skretanje robota koristi se mali kuglični kotač koji se slobodno rotira ovisno o brzini vrtnje kotača na istosmjernim motorima. Kada jedan od motora uspori ili stane, sila koju drugi motor daje rotira robota i s njim se kuglični kotač rotira. On se montira na distance koje se stavljaju na šasiju u za to predviđene rupe kako je prikazano na Slika 11.



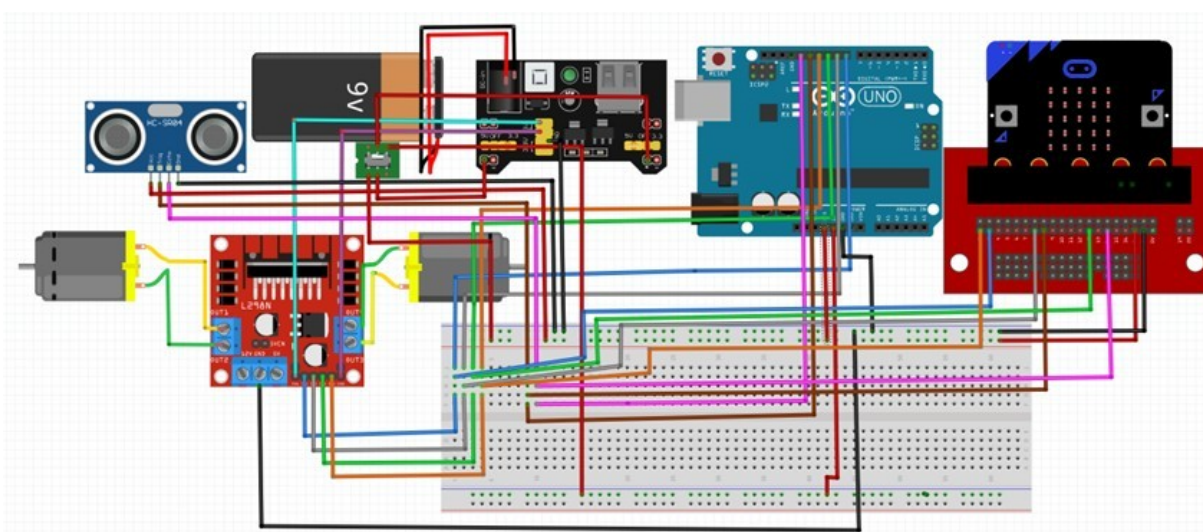
Slika 11. Montaža kugličnog kotača na šasiju

Sve komponente se nakon toga po želji montiraju na šasiju robota tako da mogu biti lako dostupne za spajanje kabela u njih. Budući da Arduino i eksperimentalna pločica imaju na

sebi ženske spojnice, a MircoBit i senzori muške, potrebno je koristiti tri vrste kabela:

- Muški – muški kabel dupont
- Muški – ženski kabel dupont
- Ženski – ženski kabel dupont.

Sve komponente se spajaju po shemi prikazanoj na Slika 12 i u Tablica 1.



Slika 12. Shema spajanja svih komponenti multiplatformskog robota

Tablica 1. Tablica pinova i spojeva

SENZOR	ARDUINO	MICROBIT	NAPAJANJE
VCC	3.3V	3.3V	3.3V
TRIG	D12	P8	N/C
ECHO	D13	P14	N/C
GND	GND	GND	GND
ENA	N/C	N/C	5V
ENB	N/C	N/C	5V
IN1	D8	P0	N/C
IN2	D9	P1	N/C
IN3	D10	P2	N/C
IN4	D11	P3	N/C
N/C	5V	N/C	5 V preko DPDT
N/C	3.3V	3.3V	3.3 V preko DPDT

Kazalo boja:

ULTRAZVUČNI SENZOR HC-SR04

H-MOST MODUL L298N

5. ZAKLJUČAK

Ovaj rad predstavlja koncept multiplatformskog robota kao inovativno rješenje za unapređenje nastave Informatike u osnovnim školama. S obzirom na to da se robot temelji na dvije različite platforme, on omogućava učenicima razumijevanje, analizu i praktično usvajanje različitih programskih jezika i koncepata programiranja te istovremeno razvija tehničke i analitičke vještine i potiče razvoj apstraktnog mišljenja. Kroz ovaj projekt učenici mogu povezati teorijska znanja Informatike i Tehničke kulture s praktičnom primjenom, čime se jača njihovo razumijevanje i motivacija za daljnje istraživanje STEM područja. Projekt ima širi edukativni potencijal jer omogućuje integraciju Informatike s ostalim nastavnim predmetima i izvannastavnim aktivnostima u osnovnoškolskom kurikulumu. Povezivanje s Tehničkom kulturom očituje se kroz zadatke sastavljanja robota, čitanja elektroničkih shema i korištenja univerzalnih pločica za spajanje svih komponenti. Moguće je i povezivanje s nastavom Fizike kroz razumijevanje principa rada istosmjernih motora i senzora, budući da se u Fizici uči električna energija i valovi. Takva interdisciplinarnost potiče učenike na kreativno razmišljanje i stvara želju za učenjem i otkrivanjem novih načina kako povezati više znanosti u funkcionalnu tvorevinu od samih početaka. Izradom ovog projekta može se realizirati i suradnja različitih dobnih skupina učenika, što razvija i poboljšava znanje rada u timskim skupinama. Microbit omogućuje početnicima učenje osnova programiranja putem blokovskog sučelja, koji je dio kurikuluma Informatike u razrednoj nastavi, te mogućnošću pregleda i uređivanja istog koda u Pythonu, koji je dio kurikuluma predmetne nastave Informatike, dok Arduino pruža platformu za osnovno i naprednije učenje programiranja u jeziku C koji je dio kurikuluma u srednjim tehničkim školama. Ovakav pristup učenicima nudi kontinuitet u učenju jer omogućuje prelazak s osnovnih na složenije koncepte programiranja bez prekida u procesu. Učenici mogu prepoznati sličnosti i razlike između različitih programskih jezika i time povezati prethodno znanje s novim iskustvom. Iako projekt nije implementiran u nastavi, potencijal za njegovu primjenu je velik. Ovakvi roboti mogli bi postati dio standardne opreme u školama, omogućujući nastavnicima da obogate nastavu Informatike i Tehničke kulture. Također, prilagodba zadataka za različiti uzrast učenika omogućila bi široku primjenu ovog robota po dobnim skupinama, od osnovnih zadataka u nižim razredima do složenih u višim. Oprema za izradu robota je povoljna i lako dostupna te je time osigurana ekonomičnost i održivost ovog rješenja. U budućnosti se ovakav projekt može dodatno razvijati dodavanjem novih funkcija i tehnologija poput primjene naprednijih senzora, komunikacijskih modula za međusobno povezivanje dvaju ili više robota, ili implementacije umjetne inteligencije. Na taj

način projekt bi mogao poslužiti ne samo za edukaciju u osnovnim školama, već i kao uvod u naprednu tehnologiju i robotiku u srednjoškolskom obrazovanju. Uz daljnju prilagodbu školskim kurikulumima ovaj projekt može postati temelj za napredniju i suvremeniju nastavu Informatike koja doprinosi učenicima i društvu.

LITERATURA

1. Arduino, Arduino Documentation, <https://docs.arduino.cc/> (pristupljeno 27. 09. 2024.)
2. AustinS89, HC-SR04 Ultrasonic Sensor + Arduino UNO, <https://instructables.com/HC-SR04-Ultrasonic-Sensor-Arduino-UNO-a-Tutorial/> (pristupljeno 1. 10. 2024.)
3. Micro:bit Educational Foundation, (2023), micro:bit Documentation, <https://microbit.org/get-started/> (pristupljeno 28. 09. 2024.)
4. Phenoptix, Distance Sensing With the Microbit and Sonar (HC-SR04 Module), <https://www.instructables.com/Distance-Sensing-With-the-Microbit-and-Sonar-HC-SR/> (pristupljeno 2. 10. 2024.)
5. Tronixlabs, Control DC and stepper motors with L298N Dual Motor Driver, Instructables, <https://www.instructables.com/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/> (pristupljeno 1.10. 2024.)
6. Wikipedia, Micro:bit, <https://en.wikipedia.org/wiki/Micro:bit> (pristupljeno 28. 09. 2024.)
7. Wikipedia, Arduino, <https://en.wikipedia.org/wiki/Arduino> (pristupljeno 27. 09. 2024.)

SAŽETAK

Ovaj diplomski rad prikazuje koncept, kreiranje i implementaciju multiplatformskog robota temeljen na Arduino i Microbit tehnologijama s ciljem unapređenja nastave Informatike u osnovnim školama. Robot omogućuje učenicima učenje i usporedbu različitih programskih jezika te povezivanje Informatike s predmetima poput Tehničke kulture i Fizike. Korištenjem komponenata poput H-mostova, istosmjernih motora i ultrazvučnih senzora, projekt učenicima pruža praktične vještine i razumijevanje interdisciplinarnih koncepata, potičući kreativnost i interes za STEM područja.

Ključne riječi: Arduino, Microbit, programiranje, informatika, učenje, robot.

SUMMARY

This thesis presents the concept and implementation of a multiplatform robot based on Arduino and microbit technologies, aimed at enhancing computer science education in primary schools. The robot enables students to learn and compare different programming languages while connecting computer science with subjects like technical culture and physics. By using components such as H-bridges, DC motors, and ultrasonic sensors, the project provides students with practical skills and interdisciplinary knowledge, fostering creativity and interest in STEM fields .

Keywords: Arduino, Microbit, programming, computer science, learning, robot.