



ŠKOLE ZA 21. VIJEK

PRIRUČNIK ZA NASTAVNIKE

U SVIJETU KODIRANJA

V, VI, VII I VIII RAZRED



Zavod za udžbenike i nastavna sredstva
PODGORICA

PROGRAMIRANJE PRIMJENOM MICRO:BIT UREĐAJA ZA PETI, ŠESTI, SEDMI I OSMI RAZRED OSNOVNE ŠKOLE

Priručnik za nastavnike

Autori: dr Goran Šuković, mr Dušanka Vujičić, Biljana Krivokapić, Zineta Husić, Katarina Veljković i Nebojša Vasiljević



Zavod za udžbenike i nastavna sredstva
PODGORICA, 2022.

Autori: dr Goran Šuković, mr Dušanka Vujičić, Biljana Krivokapić, Zineta Husić, Katarina Veljković i Nebojša Vasiljević

Izdavač: British Council

Suizdavač: Zavod za udžbenike i nastavna sredstva – Podgorica

Za izdavača: Vanja Madžgalj, direktorica

Za suizdavača: mr Aleksandra Hajduković, v.d. direktorice

Glavni urednik: Radule Novović

Odgovorni urednik: Lazo Leković

Urednica izdanja: Vanja Madžgalj

Recenzenti: dr Milenko Mosurović, dr Božo Krstajić, Radovan Sredanović, Nađa Luteršek i mr Zoran Lalović

Lektura i korektura: Porta Aperta

Grafičko oblikovanje: Sanjin & Friends

Prelom: Sanjin & Friends

Tehnički urednik: Dajana Vukčević

Fotografije: British Council

Štampa: DPC Podgorica

Tiraž: 300

CIP – Каталогизacija u publikaciji
Nacionalna biblioteka Crne Gore, Cetiње
ISBN 978-86-907231-2-6 (British Council)
ISBN 978-86-303-2431-4 (ZUNS)
COBISS.CG-ID 22932996

Nacionalni savjet za obrazovanje, rješenjem br. 19-05-119/22-6418/3 od 29. jula 2022. godine, odobrio je ovaj priručnik za upotrebu.

Copyright © British Council 2022

Zavod za udžbenike i nastavna sredstva – Podgorica je nosilac međunarodne licence Creative Commons Attribution Non-Commercial ShareAlike 4.0 International Licence (CC BY-NC-SA 4.0) za korišćenje ove publikacije.

©British Council 2022

Ulcinjska 8

81000 Podgorica

Crna Gora

Ovaj priručnik nastao je u okviru obrazovnog programa „Škole za 21. vijek”, koji sprovodi British Council u saradnji sa Ministarstvom prosvjete i Zavodom za školstvo Crne Gore.

Sadržaj

PREDGOVOR	4
UVOD	7
O MICRO:BIT-U	8
V RAZRED	11
OBRAZOVNO-VASPIITNI ISHOD 6	12
DIDAKTIČKO-METODIČKA UPUTSTVA	13
ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE	13
PRAĆENJE I PROCJENJIVANJE OBRAZOVNIH POSTIGNUĆA	50
PROJEKTNNA NASTAVA	51
RAD NA PROJEKTU – „MJERENJE TEMPERATURE NA DALJINU“	51
RAD NA PROJEKTU – „POTAPANJE BRODOVA“	52
RAD NA PROJEKTU – „PAZI, LOPOVI!“	55
OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA	56
VI RAZRED	59
OBRAZOVNO-VASPIITNI ISHOD 5	60
DIDAKTIČKO-METODIČKA UPUTSTVA	61
ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE	61
PRIMJERI PRAĆENJA PROCJENJIVANJA	84
PROJEKTNNA NASTAVA	85
RAD NA PROJEKTU – „ULTRAZVUČNI SENZOR“	85
RAD NA PROJEKTU – „SEMAFOR“	88
OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA	90
VII RAZRED	93
OBRAZOVNO-VASPIITNI ISHOD 5	94
DIDAKTIČKO-METODIČKA UPUTSTVA	95
ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE	95
PRIMJERI PRAĆENJA PROCJENJIVANJA	117
PROJEKTNNA NASTAVA	118
OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA	118
VIII RAZRED	121
OBRAZOVNO-VASPIITNI ISHOD 8	122
DIDAKTIČKO-METODIČKA UPUTSTVA	123
ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE	123
PRIMJERI PRAĆENJA PROCJENJIVANJA	153
PROJEKTNNA NASTAVA	153
OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA	154
KLJUČNI POJMOVI PREMA REDOSLJEDU POJAVLJIVANJA	154
LITERATURA	157
DODATAK 1 – PRILOG ZA ISHOD „SORTIRANJE NIZA SELEKCIJOM“	157
LISTA AKTIVNOSTI ZA ISHOD „SORTIRANJE SELEKCIJOM“	157
DODATAK 2 – PRILOG ZA PROJEKAT „SIMULACIJA SLUČAJNIH DOGAĐAJA“	187
SIMULACIJA SLUČAJNIH DOGAĐAJA	187

PREDGOVOR

Ovaj priručnik za nastavnike nastao je u okviru programa „Škole za 21. vijek”, koji je dizajnirao i sprovodi Britanski savjet (British Council) širom regiona Zapadnog Balkana u periodu od 2019. do 2022. godine kroz finansijsku podršku britanske vlade.

Program je nastao kao odgovor na dominantno prisustvo novih tehnologija u savremenoj ekonomiji i potrebu da se obrazovni sistemi ovog regiona usklade sa zahtjevima tržišta rada, koje zahtijeva visok stepen digitalne pismenosti i vještina koje podstiču razvoj inovacija. Crna Gora je u svojoj novijoj ekonomskoj strategiji prepoznala IT industriju i inovacije kao model diversifikacije i rasta ekonomije, te pokrenula proces digitalne transformacije, koja ima za cilj da izgradi potrebnu infrastrukturu i podstakne razvoj kadrova koji će transformisati biznis sektor i učiniti crnogorsku privredu relevantnim igračem na sve konkurentnijem globalnom tržištu.

Program „Škole za 21. vijek” polazi od pretpostavke da su ljudi najveći resurs ekonomije svakog društva i da kapacitet neke industrije ili biznisa da rješava izazove, razvija se i raste u velikoj mjeri zavisi od stepena kreativnosti, fleksibilnosti i vještina rješavanja problema njenih ljudi. Relevantne ekonomske analize upućuju na to da tzv. meke vještine, među kojima je kreativnost na prvom mjestu, idu ruku pod ruku sa tehničkim vještinama poput programiranja i, na neki način, čine neodvojivu cjelinu. Razvoj inovacija bilo koje privredne grane i integrisanje digitalnih tehnologija u prostore poput javnih komunikacija, obrazovanja i kulture, nezamislivi su bez snažnog upliva kreativnosti i umijeća analize problema kroz kritičku diskusiju. Stoga, podsticanje kreativnosti od najranijeg doba u nastavnom procesu, u kombinaciji sa različitim digitalnim vještinama, preduslov je za stvaranje kadra za ekonomiju 21. vijeka.

Kodiranje Micro:bit uređajem, kao ključni segment programa „Škole za 21. vijek”, predstavlja jedinstvenu platformu za razvoj kreativnosti, inovacija i rješavanje problema na jednom mjestu kroz rad na inovativnim projektima. BBC Micro:bit je relativno nova tehnologija nastala 2016. godine u Velikoj Britaniji, kao odgovor na činjenicu da postoji ogroman jaz u digitalnim vještinama na tržištu rada. Micro:bit predstavlja mini-kompjuter sa izuzetno velikim potencijalom, koji je posebno osmišljen za razvoj bazičnih vještina programiranja kod djece u ranom uzrastu i stoga je idealan za primjenu u osnovnim školama.

Projekat „Škole za 21. vijek” uzima u obzir da je obrazovni proces složen ekosistem činilaca koji moraju djelovati jedinstveno, te da je potreban integrativan i sistemski pristup ako se želi postići kvalitetna, temeljna i dugoročna promjena. Stoga, ovaj program uključuje niz aktivnosti koje obezbjeđuju dobru povezanost i koherentnu nadgradnju različitih elemenata sistema poput kurikuluma, profesionalnog razvoja nastavnika i školskih lidera, nastavnih materijala, školske IT infrastrukture, kontrole kvaliteta i vannastavnih aktivnosti za učenike. U okviru ovog programa, i zahvaljujući dodatnoj finansijskoj podršci Ministarstva prosvjete, koja je omogućila proširenje projekta, izvršena je obuka kompletnog nastavnog kadra u osnovnim školama, uključujući preko 5.000 nastavnica i nastavnika predmetne i razredne nastave, kao i direktora 162 osnovne škole, u oblasti nastave kritičkog mišljenja, rješavanja problema i kodiranja Micro:bit-om. Donirano je preko 4.000 Micro:bit uređaja sa dodatnom opremom koja sadrži oko 50 različitih komponenti kako bi svaka škola mogla osnovati klub za kodiranje i ponuditi učenicima nove mogućnosti za razvoj primijenjenih digitalnih vještina.

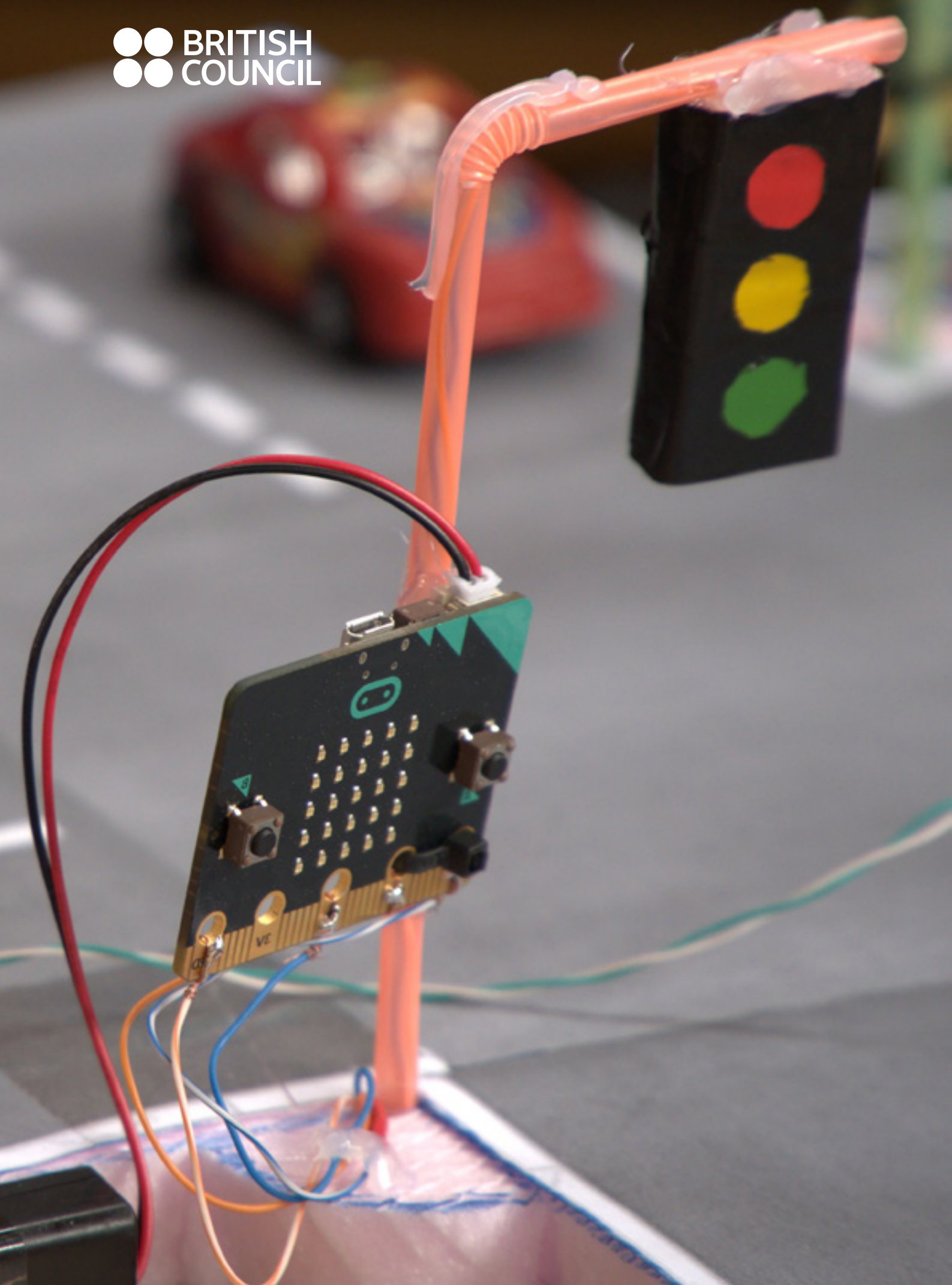
Inoviran je predmetni program Informatika sa tehnikom za V, VI, VII i VIII razred, čime je uvedeno kodiranje kao novi segment kurikuluma. Posebna dodatna obuka u kodiranju sprovedena je za preko 200 nastavnika predmeta Informatika sa tehnikom kako bi se upoznali sa novim predmetnim programom i kako bi se obezbijedila dodatna mentorska podrška IT industrije, u cilju osnaživanja nastavnika u radu sa novim tehnologijama.

Izrada priručnika za nastavnike „U svijetu kodiranja” predstavlja i završni korak u procesu inoviranja nastave, sa ciljem da se pruži kvalitetna podrška glavnim nosiocima procesa digitalne transformacije obrazovanja i obezbijedi konzistentan pristup nastavi kodiranja. Priručnik je koncipiran tako da prati obrazovno-vaspitne ishode vezane za razvoj vještina kodiranja predmetnog programa za V, VI, VII i VIII razred, i predstavlja važnu dopunu postojećeg udžbenika „Informatika sa tehnikom”.

Ovim je stvorena snažna osnova za razvoj digitalnih vještina kod učenika u osnovnim školama, ali i, što je jednako važno, osnova za razvoj kreativnosti, inovativnosti, kritičkog mišljenja i rješavanja problema.

Uključivanjem svih osnovnih škola i područnih odjeljenja u Crnoj Gori u ovaj program, željeli smo da stvorimo inkluzivno okruženje za razvoj i napredak svakog djeteta u zemlji, a posebno djevojčica, i time damo doprinos budućem prosperitetu crnogorskog društva.

Vanja Madžgalj
Direktorica Britanskog savjeta u Crnoj Gori



UVOD

Ovaj priručnik nastao je s ciljem da nastavniku¹ pruži smjernice za realizaciju obrazovno-vaspitnih ishoda koji se odnose na programiranje primjenom Micro:bit uređaja, a sastavni su dio kurikuluma. U ovom priručniku naizmjenično koristimo termine „programiranje”, kao širi pojam razvoja softvera, i „kodiranje”, kao dio programiranja koji koristi jednostavne kodove kao početne korake u procesu programiranja.

Priručnik podstiče nastavnike da putem raznovrsnih aktivnosti učenja kod učenika razvijaju kritičko mišljenje, kreativnost, inicijativu, samopouzdanje, odgovornost, preciznost, timski rad, kao i vještine programiranja i rješavanja problema. Tokom procesa nastave treba se bazirati na praktičnoj primjeni stečenih znanja.

Kako se predmet Informatika sa tehnikom izučava u petom, šestom, sedmom i osmom razredu i kako se izučavanje programiranja upotrebom Micro:bit-a prožima po vertikali, tako je i sam priručnik koncipiran da odgovori potrebama obrazovno-vaspitnih ishoda zastupljenih u sva četiri razreda. Nastavniku je ponuđen određeni broj didaktičko-metodičkih rješenja kao podrška i inspiracija u kreiranju nastavnog procesa u dijelu ishoda namijenjenih programiranju i primjeni Micro:bit uređaja sa pratećom opremom u redovnom nastavnom procesu.

Priručnik sadrži opšte informacije o dijelu predmetnog programa Informatika sa tehnikom, koji se odnosi na kodiranje i projektu „Škole za 21. vijek”. Značajan dio Priručnika posvećen je ciljevima, obrazovnim ishodima i ishodima učenja, mehanizmima za procjenu postignuća učenika, kao i didaktičko-metodičkim uputstvima za organizaciju, planiranje i realizaciju nastave. Sastavni dio Priručnika su izrađeni primjeri priprema za nastavu u odnosu na konkretne obrazovno-vaspitne ishode, koji se odnose na programiranje u petom, šestom, sedmom i osmom razredu.

Podrazumijeva se da će nastavnik od ponuđenog materijala iskoristiti ono što smatra da mu je potrebno, a da će prilikom pripreme scenarija časa unijeti i svoj pristup i postupke, kojima izražava odnos prema sadržajima koje će usvojiti učenici.

U okviru cjeline „Časovi programiranja“ izrađeni su primjeri scenarija časa za sve ishode koji se odnose na programiranje u petom, šestom, sedmom i osmom razredu predmeta Informatika sa tehnikom.

Važan segment procesa učenja kodiranja jesu praktične vježbe i projektni rad, tokom kojih učenici ovladavaju primjenom kodiranja u fizičkom prostoru i sintezom različitih elemenata koji čine finalni proizvod, kao što su dizajn makete, povezivanje različitih komponenti sa Micro:bit-om i testiranje funkcionalnosti. Stoga ovaj priručnik nudi niz primjera za izradu zadataka koji zahtijevaju praktičnu primjenu naučenog.

Kriterijumi za praćenje postignuća obrazovno-vaspitnih ishoda i ocjenjivanje učenika podijeljeni su u tri nivoa: osnovni, srednji i viši.

Izrađeni su primjeri nastavnih listića, koje nastavnici mogu koristiti u potpunosti ili u djelovima za koje smatraju da najbolje odgovaraju učenicima u skladu sa njihovim uslovima rada.

Prijedlozi scenarija časova omogućavaju nastavniku, čija je uloga prije svega mentorska, a ne predavačka, jer pomaže i usmjerava rad učenika, da organizuje rad sa učenicima kroz njihovu aktivnost i aktivne strategije učenja.

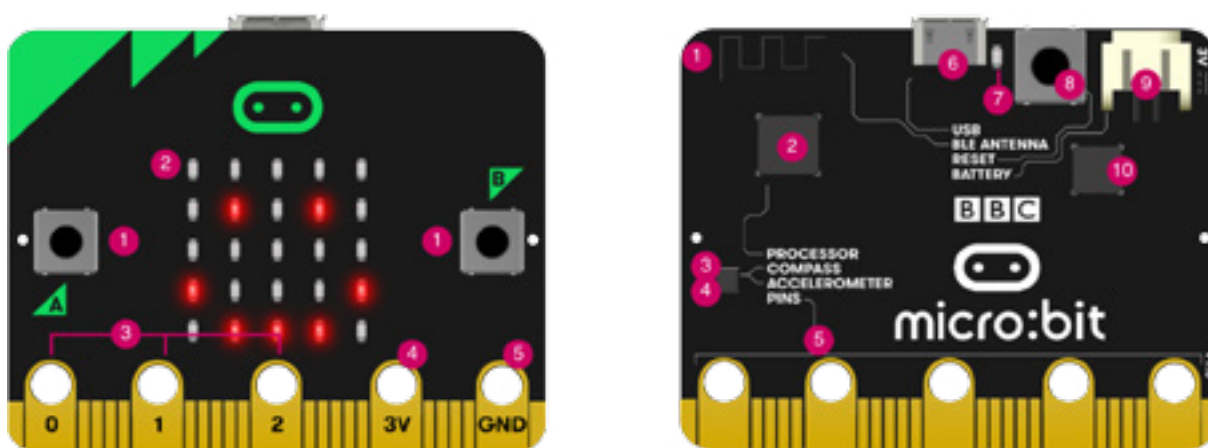
¹ Svi izrazi koji se u ovom programu koriste u muškom rodu obuhvataju iste izraze u ženskom rodu.

O MICRO:BIT-U

Micro:bit je ručni, programabilni mikroračunar koji je razvijen prvenstveno kako bi zainteresovao djecu za programiranje. Za kreiranje programa koristi se razvojno okruženje MakeCode. Zavisno od tehničkih uslova, razvojno okruženje MakeCode se može koristiti *online* ili *offline*.

„Mozak” Micro:bit-a je ARM procesor, a osnovnu komunikaciju sa korisnikom omogućavaju tasteri i svijetleće diode, koji se nalaze na uređaju. Osim toga, Micro:bit je opremljen i brojnim sensorima koji će nam omogućiti razne zanimljive primjene ovog mikroračunara.

Ekran



Na prednjoj strani Micro:bit-a nalazi se 25 lampica koje su poređane u pet vrsta i pet kolona. Ove lampice izrađene su u takozvanoj LED tehnologiji. LED je skraćenica od Light Emitting Diode, što u prevodu znači „dioda koja emituje svjetlost, svijetleća dioda”. Rasvjeta napravljena ovom tehnologijom, zahvaljujući dugotrajnosti i energetske efikasnosti, sve više zamjenjuje halogenu i fluorescentnu rasvjetu u domaćinstvima.

Tasteri

Sa lijeve i desne strane ekrana nalazi se po jedan taster. Tasteri su označeni slovima A i B i koriste se kao ulazi. Micro:bit može da registruje koji taster je pritisnut i da reaguje na pritisak nekog od njih tako što će izvršiti neku radnju ili poslati informaciju nekom drugom uređaju, zavisno od toga kako ga programiramo.

Na poledini Micro:bit-a nalazi se još jedan taster koji omogućava ponovno pokretanje (resetovanje) učitano programa.

Kompas

Micro:bit ima ugrađen i kompas. Možemo da odredimo jačinu magnetnog polja u njegovoj blizini, kao i trenutnu orijentaciju uređaja u odnosu na strane svijeta.

Akcelerometar

Akcelerometar nam omogućava da detektujemo različite tipove pokreta Micro:bit-a. Možemo registrovati kada ga neko protrese, nagne, ispusti. Ovaj senzor svakako može doprinijeti da naši projekti postanu sadržajni i još interesantniji.

Senzor za temperaturu

Ne postoji poseban senzor za temperaturu koji je ugrađen u Micro:bit, ali je omogućeno da koristimo senzor koji mjeri temperaturu njegovog procesora. Temperatura koju pokazuje Micro:bit nije potpuno precizna, ali možemo da registrujemo njene promjene i napravimo različite projekte koji koriste te podatke.

Pinovi

Na ivici Micro:bit-a nalazi se 25 izvoda (pinova) – 5 velikih i 20 malih. Pet velikih (osnovnih) pinova označeni su sa 0, 1, 2, 3V i GND. Pinovi 0, 1 i 2 su ulazno-izlazni pinovi opšte namjene. Preko njih, upotrebom provodnika sa krokodil-štikaljkama, možemo na Micro:bit povezati razne dodatne senzore, ali i upravljati brojnim uređajima. Pinovi 3V i GND (*ground* – „uzemljenje, masa“) služe za napajanje. Iako je obilježen sa 3V, na izlazu ovog pina dobijamo napon od 3,3V u odnosu na masu.

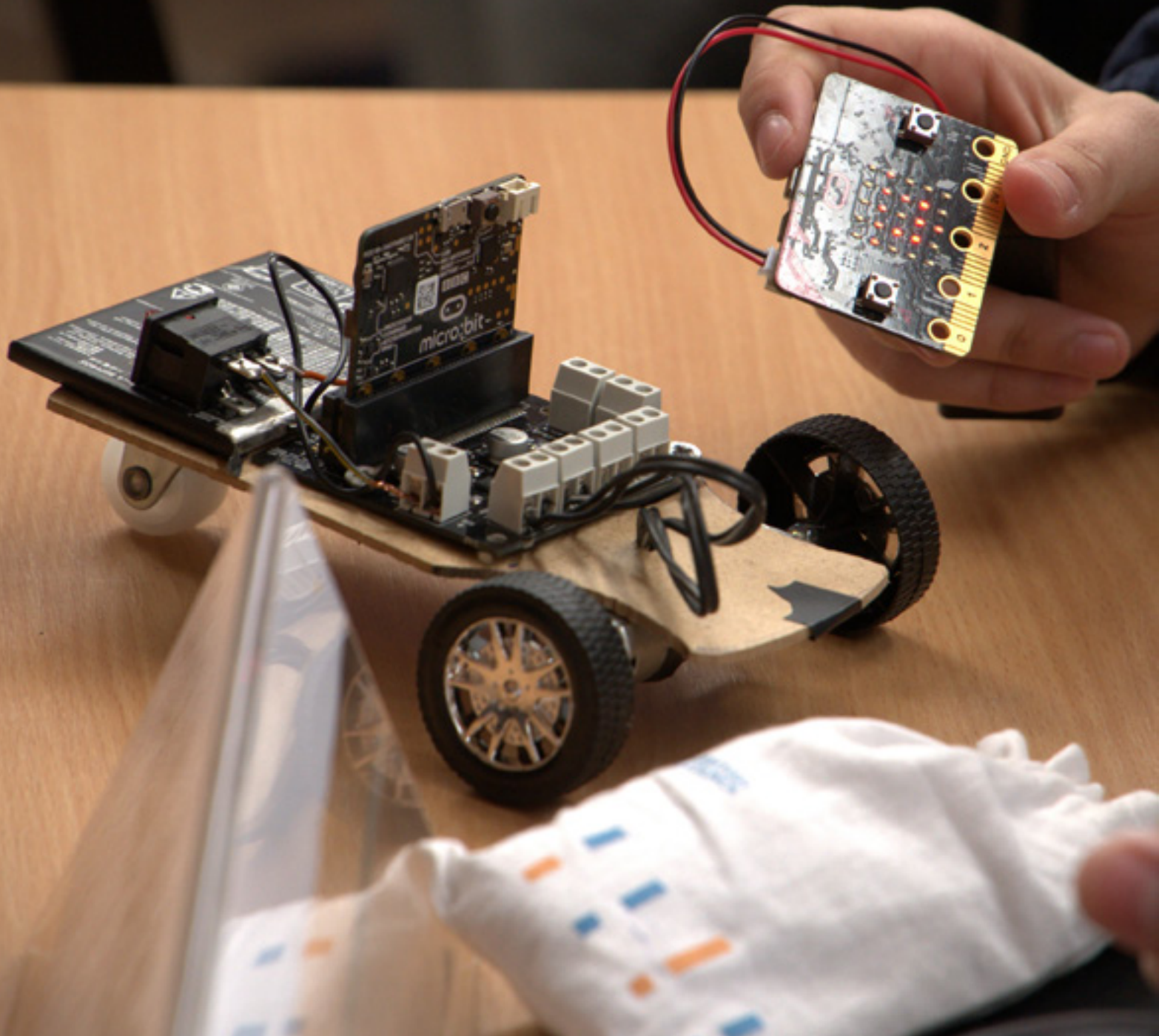
Bluetooth

Micro:bit je opremljen i BLE (Bluetooth Low Energy) antenom, pomoću koje možemo bežično da komuniciramo sa više drugih Micro:bit-ova, računarom, mobilnim telefonom ili nekim drugim uređajem. Komunikacija sa drugim uređajima je dvosmjerna, što znači da Micro:bit može da prima podatke sa njih, ali i da im prosljeđuje podatke.

Micro:bit se najjednostavnije programira preko uređivača blokova, ali nije mnogo složeno ni ako programiramo u nekom od pravih, tekstualnih programskih jezika, poput Javaskripta ili Pajtona.

Verzija Pajtona koja se koristi u radu sa Micro:bit-om (i na nekim drugim, sličnim platformama) zove se *MicroPython*. Mikropajton je pojednostavljena verzija Pajtona, koja ne podržava baš sve što postoji u standardnom Pajtonu. Na primjer, u Mikropajtonu nećemo moći da koristimo naša slova i ostale Unicode karaktere ili da formatiramo stringove. Moguće je da ćete naići na još poneko ograničenje, ali šanse za to su male jer ćete imati na raspolaganju sve dijelove Pajtona koji nama mogu zatrebati.

Okruženje u kome se Micro:bit najjednostavnije programira u Pajtonu (tačnije Mikropajtonu) je Mu (Mju). Iako ćemo ovo okruženje najviše koristiti samo kao editor za pisanje programa, ono u sebi sadrži i druge korisne mogućnosti.





V RAZRED

OBRAZOVNO-VASPITNI ISHOD 6

Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.

Ishodi učenja

Po završetku učenja učenik će moći da:

- demonstrira metod transformisanja pripremljenih programa u fajlove za izvršavanje;
- razumije da računar uzima ulazne podatke i transformiše ih u izlazne;
- upotrebljava različite tipove informacija kao ulaz za računar;
- izvršava programe na uređaju koji uzimaju ulaz i proizvode izlaz.

Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda

a) Sadržaji/pojmovi:

Naglasak je na primjerima transformacija već pripremljenih programa za uređaje u fajlove za izvršavanje, prenošenju programa na uređaj i izvršavanju programa, upoznavanju sa svim tipovima ulaza za raspoložive tipove uređaja (npr. Micro:bit) i promjeni parametara datih programa. Pojmovi: ulaz, izlaz, ekran (engl. *display*), senzori, program, radno okruženje za razvoj programa.

Napomena: Nije potrebno objašnjavati detalje hardvera uređaja, detalje o pojedinačnim naredbama, niti pojam algoritma.

b) Aktivnosti učenja

Učenici:

- vježbaju prenošenje gotovih programa na uređaj (Micro:bit, Raspberry PI, Lego robot...) i izvršavaju ih;
- na sliku uređaja lijepe stikere koji pokazuju hardverske komponente;
- crtaju tabelu koja opisuje tipove ulaza za dati uređaj, dugme, senzori, brojevi, tekst...;
- vježbaju upotrebu radnog okruženja za razvoj programa;
- izvršavaju programe koji koriste više vrsta ulaza; npr. oba dugmeta, vrijednosti senzora, ruke Lego robota...;
- mijenjaju djelove programa (npr. dužinu trajanja zvučnog signala) i posmatraju rezultate izmjena.

c) Broj časova realizacije (okvirno) – 7 časova.

*OBRAZOVNO-VASPITNI ISHOD 6 preuzet iz Plana i programa predmeta Informatika sa tehnikom.



DIDAKTIČKO-METODIČKA UPUTSTVA

Opšte didaktičke preporuke

Akcentat ovoga predmeta, sa posebnim osvrtom na ishode koji se odnose na programiranje, stavlja se na praktično znanje, pa je suština aktivnosti u učenju bazirana na praktičnim vježbama uz asistenciju nastavnika. U skladu sa dinamikom razvoja tehnologije i izvjesnih tehničkih inovacija, planirani nastavni sadržaji ovoga predmeta biće podložni izmjenama i dopunama. Zato ovaj priručnik, u njegovoj realizaciji, ostavlja slobodu svim učesnicima procesa. Predviđeni sadržaji zahtijevaju raznovrstan metodički pristup, tj. obuhvataju korišćenje određenog programskog jezika, upotrebu raznih uređaja, upoznavanje i upravljanje raznim komponentama – ultrazvučni senzori, relej, servo motor, senzor pokreta, DC motora, foto-otpornika i sl.

Didaktičke preporuke za realizaciju OVI

Obrazovno-vaspitni ishod o ulaznim i izlaznim uređajima i programima realizovati kroz praktični rad. Preporučuje se upotreba uređaja Micro:bit i drugih sličnih uređaja (Arduino, Raspberry PI, Lego robot...) ili desktop/laptop računara (npr. program napisan u jeziku C/C++ koji ima kôd). Demonstrirati načine transformacije programa u izvršne verzije, ulaz i izlaz programa (npr. kod Micro:bit uređaja dugme, gest...; kod desktop računara tastatura i ekran...) i promjenu parametara programa (npr. brzinu pojavljivanja slike na uređaju...).

*DIDAKTIČKO-METODIČKA UPUTSTVA preuzeta iz Plana i programa predmeta Informatika sa tehnikom.



ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE

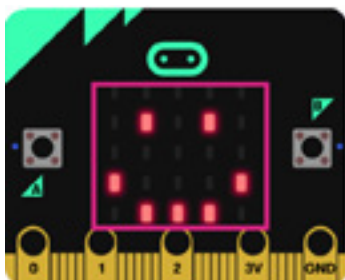
Prilikom pripreme materijala za realizaciju nastave, nastavnik treba da kreira materijale koji omogućavaju aktivno uključivanje učenika. U ovom segmentu razrađeno je 7 scenarija za časove, dat je prijedlog primjera kriterijuma ocjenjivanja po nivoima postignuća i 3 projekta.

PLAN ČASA BROJ 1	
OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da opiše Micro:bit uređaj i sve ono što on sadži.
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, internet konekcija.
KRATAK OPIS ČASA	
Učenici se upoznaju sa sadržajem Micro:bit uređaja, njegovom prednjom i zadnjom stranom i šta sve on sadrži.	

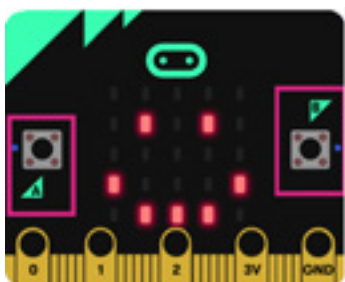
Aktivnost 1

Upoznavanje sa Micro:bit-om – Učenici se upoznaju sa Micro:bit uređajem tako što nastavnik putem prezentacije prikazuje i objašnjava svaku stavku ovog uređaja, dok učenici prate i direktno analiziraju na uređaju koji im je nastavnik prethodno dodijelio. Učenici upoznaju djelove na prednjoj strani uređaja.

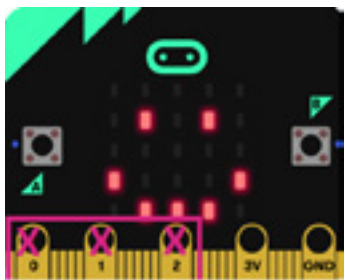
LED diode – Učenici prate izlaganje nastavnika i upoznaju se sa ulogom LED dioda (engl. *Light-emitting Diode*) i njihovom ulogom da emituju svjetlo. Na ekranu BBC-jevog Micro:bit-a ima 25 crvenih LED dioda (postavljenih u mrežu 5 x 5), koje se mogu pojedinačno programirati i koje zavisno od programa mogu prikazivati određeni tekst, brojeve, figure.



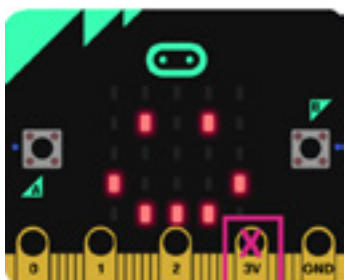
Tasteri – Učenici prate izlaganje nastavnika i upoznaju se sa tasterima A i B (dugmad koja služe za pokretanje programa), a koji predstavljaju najčešće ulazne uređaje. Pritiskom na svaki taster pojedinačno ili pritiskom na oba tastera u isto vrijeme, pokreće se program na uređaju.



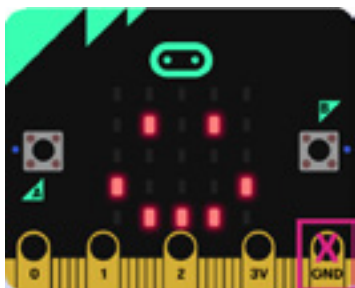
Pinovi 0, 1 i 2 – Učenici prate izlaganje nastavnika i upoznaju se sa pinovima 0, 1 i 2, koji su ulazno-izlazni pinovi opšte namjene (skraćeno GPIO). Na GPIO izvode mogu se priključiti slušalice, senzor dodira i ostala elektronika za dodatne funkcije Micro:bit-a. Preko njih možemo upravljati brojnim uređajima.



Pin 3V – Učenici prate izlaganje nastavnika i upoznaju se sa pinom 3V, koji služi za napajanje (učenici ne znaju šta znači 3V, nastavnik procjenjuje da li će upotrijebiti termin).

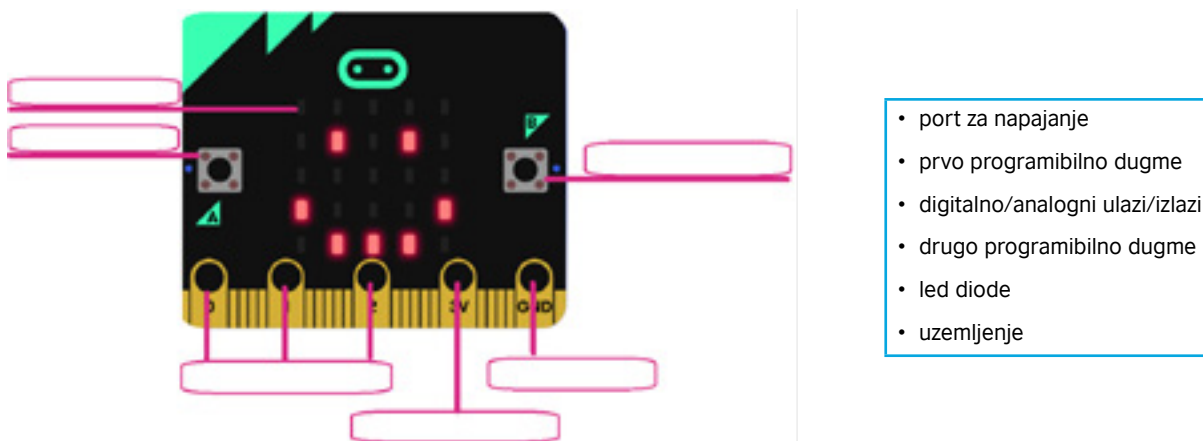


Pin GND – Učenici prate izlaganje nastavnika i upoznaju se sa pin GND (ground – „uzemljenje“), koji služi za zatvaranje strujnoga kruga kada se na Micro:bit priključe slušalice, LED diode ili spoljašnji prekidači.



Aktivnost 2

Zadatak – Učenici rješavaju kratak test na osnovu svega pomenutog tako što u prazna polja upisuju nazive dijelova iz kolone desno, a zatim provjeravaju odgovore.



Aktivnost 3

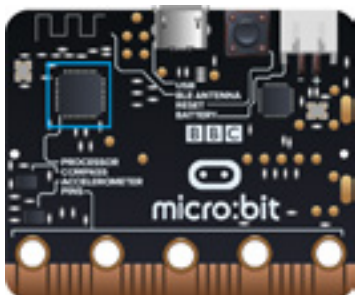
Zadnja strana Micro:bit uređaja – Učenici upoznaju dijelove na zadnjoj strani uređaja.

Radio i Bluetooth antena – Učenici prate izlaganje nastavnika i upoznaju se sa radio i Bluetooth antenama. Komponenta za radio omogućava bežično komuniciranje između dva ili više Micro:bit-ova. Radio možemo koristiti za slanje poruka drugim Micro:bit-ovima, u kreiranju igara ili kvizova za više igrača i slično.

BLE (Bluetooth Low Energy) antena omogućava Micro:bit-u da šalje i prima Bluetooth signale, odnosno da bežično komunicira sa računarima, mobilnim telefonima i tabletima.



Procesor – Učenici prate izlaganje nastavnika i upoznaju se sa ulogom procesora kao Micro:bit-ovog mozga; procesor prihvata, dekodira i izvršava naredbe. On sadrži i toplotni senzor pomoću kojeg se može izmjeriti temperatura okoline.



Kompas – Učenici prate izlaganje nastavnika i upoznaju se sa ulogom kompasa kao dijela Micro:bit-a, koji detektuje Zemljino magnetno polje, što nam daje mogućnost određivanja smjera u kojem je okrenut Micro:bit. Prije upotrebe kompas treba da bude kalibrisan.



Akcelerometar – Učenici prate izlaganje nastavnika i upoznaju se sa mjerачem ubrzanja. Akcelerometar je komponenta koja registruje pokrete, tj. promjenu u brzini i položaju, odnosno moguće je izmjeriti brzinu kretanja. Pomoću ove komponente moguće je registrovati kada se Micro:bit pomjeri. Takođe, moguće je detektovati i druge aktivnosti, na primjer drhtanje, nagib ili slobodan pad.



Samostalna žuta LED lampica – Učenici prate izlaganje nastavnika i upoznaju se sa ulogom samostalne žute LED lampice na zadnjoj strani Micro:bit-a. Primjećuju da ova lampica svijetli pri preuzimanju programa i pri napajanju preko USB utičnice.



Taster za resetovanje – Učenici prate izlaganje nastavnika i zaključuju da se na poledini Micro:bit-a nalazi taster koji omogućava ponovno pokretanje (resetovanje) učitanoog programa.



Utičnica (konektor, priključak) za baterije – Učenici prate izlaganje nastavnika i zaključuju da, ako ne žele da se Micro:bit napaja preko USB utičnice, mogu ga isključiti sa računara i spojiti na držač za baterije. Ovo je praktično ako žele Micro:bit koristiti napolju ili igrati igrice na njemu. Samo dvije AAA baterije mogu ga veoma dugo napajati.



Čip – Učenici prate izlaganje nastavnika i saznaju da čip upravlja USB vezom i koristi se za učitavanje novog koda u Micro:bit, te za slanje i preuzimanje serija podataka između Micro:bit-a i računara.



Aktivnost 4

Zadatak – Učenici rješavaju kratak test na osnovu kojeg će provjeriti u kojoj mjeri su se upoznali sa djelovima Micro:bit-a koji se nalaze na zadnjoj strani. Njihov zadatak je da na linijama napišu nazive dijelova Micro:bit-a.

mikro USB konektor

akcelerometar

dugme za resetovanje

kompas

procesor

radio i Bluetooth antena

žuta LED lampica

utičnica za baterije

čip

1 _____

2 _____

3 _____

4 _____



5 _____

6 _____

7 _____

8 _____

9 _____

Test za učenike:

Provjeri koliko znaš – zaokruži tačne odgovore:

1. Koliko LED dioda se nalazi na prednjoj strani Micro:bit-a?
 - A. 15
 - B. 21
 - C. 23
 - D. 25

2. Koja komponenta Micro:bit-a omogućava detekciju pokreta?
 - A. Kompas
 - B. Akcelerometar
 - C. Ram
 - D. Bluetooth

3. Koja je uloga kompasa na Micro:bit-u?
 - A. Detektuje svjetlost
 - B. Detektuje temperaturu
 - C. Detektuje jačinu magnetnog polja
 - D. Detektuje pokret

4. Koliko tastera ima Micro:bit?
 - A. 2
 - B. 1
 - C. 3
 - D. 4

5. Koja je uloga LED dioda na Micro:bit-u?

- A. Emituje svjetlost
- B. Detektuje temperaturu
- C. Detektuje promjenu brzine
- D. Detektuje jačinu magnetnog polja

6. (Za dodatnu nastavu) Ako znaš da je 3V pozitivan, a GND negativan pin, i označimo ih kao na ovoj slici,



koja slika ispod će pokazati tačan smjer struje?



TAČNI ODGOVORI

1 – D

2 – B

3 – C

4 – C

5 – A

6 – B

PLAN ČASA BROJ 2

OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da opiše Microsoft MakeCode okruženje i način pisanja kôda slaganjem blokova.
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, mikro USB kabl, internet konekcija.

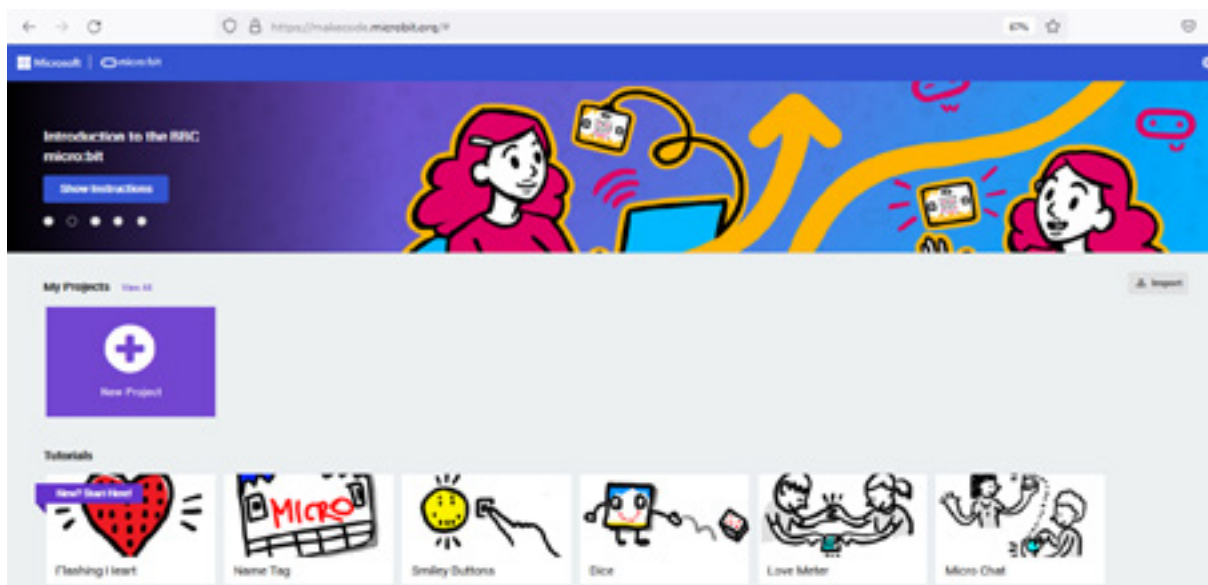
KRATAK OPIS ČASA

Kroz niz aktivnosti učenici se upoznaju sa Microsoft MakeCode okruženjem. Upoznaju funkcije blokova iz kategorije *Basic* i kreiraju svoj prvi kôd. Uočavaju da do rješenja problema dolaze korak po korak. Nakon uspješno kreiranog kôda, učenici će upoznati način na koji će kôd preuzeti i pokrenuti na Micro:bit-u.

Aktivnost 1

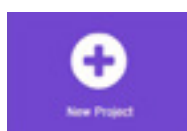
Upoznavanje sa Microsoft MakeCode okruženjem – Učenici se upoznaju sa Microsoft MakeCode okruženjem tako što prate rad nastavnika koji putem direktne demonstracije prikazuje i objašnjava djelove radne površine. Učenici, uz uputstva nastavnika, u pretraživaču upisuju internet adresu <http://www.makecode.com> na kojoj se nalazi programsko okruženje MakeCode.

Učenici uočavaju da je prednost ovog okruženja za programiranje u tome što nije potrebna instalacija programa na računar. Dovoljno je da na računaru imamo instaliran bilo koji veb pretraživač i konekciju na internet. Druga prednost je ta što ne moramo biti registrovani na sajt.

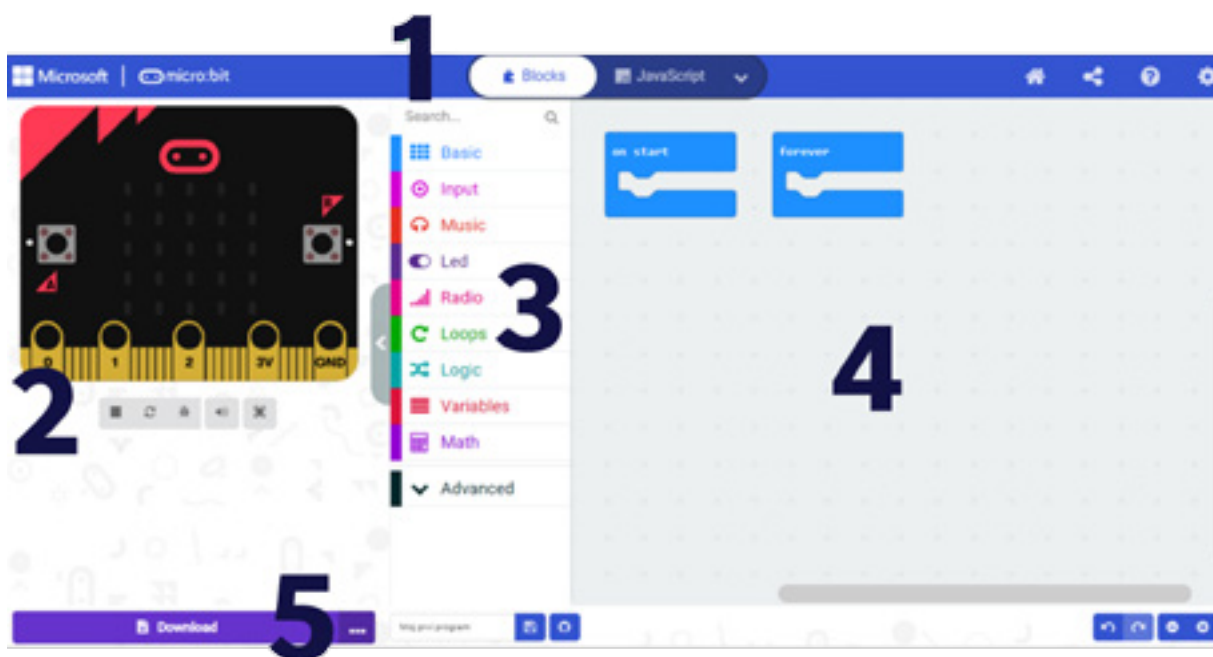


Aktivnost 2

Učenici prate rad nastavnika, otvaraju novi projekat klikom na plavo polje „New Project” i upisuju naziv projekta, npr. „Moj prvi projekat”.



Prateći izlaganje nastavnika, učenici se upoznaju sa radnim okruženjem (interfejsom). Nastavnik upoznaje učenike sa trakom menija (1) na vrhu ekrana i ukazuje na nekoliko funkcija, od kojih su najznačajnija dva centralna dugmeta koja nam omogućavaju brzo prebacivanje između vizuelnog i tekstualnog programskog jezika.



Na lijevoj strani učenici uočavaju simulator (2). Nastavnik slaže dva bloka, a učenici primjećuju da se simulator pokreće automatski svaki put kada se napravi neka promjena u programu. Dugmići ispod simulatora omogućavaju zaustavljanje/pokretanje programa i kontrolu drugih djelova simulatora (zvuka i postavljanja veličine simulatora (da li će simulator biti vidljiv preko cijelog ekrana ili ne)).

Desno od simulatora je paleta sa blokovima alatki (3), gdje se nalaze različite kategorije programskih blokova. Učenici saznaju da kategorije *Basic*, *Input*, *Music*, *Led* i *Radio* omogućavaju pristup specifičnim karakteristikama Micro:bit uređaja. Kategorije *Loops*, *Logic*, *Variables* i *Math* pružaju pristup osnovnim programskim strukturama, dok se pod kategorijom *Advanced* nalaze dodatne funkcije koje se koriste za rad Micro:bit uređaja, kao što su pinovi i drugi uređaji.

Desno od okvira sa alatkama nalazi se radna površina za programiranje – uređivač ili editor (4), na kome kreiramo/uređujemo program.

Na dnu ekrana (5) je traka sa alatkama, koja sadrži dugme za preuzimanje, koje koristimo za čuvanje programa u datoteci koja se kopira na Micro:bit uređaj. Ostale alatke na traci su za uključivanje/isključivanje zvuka i uvećavanje/smanjivanje prikaza simulatora.

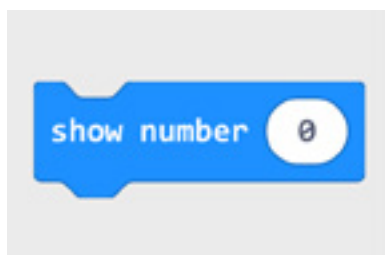
Aktivnost 3

Učenici prate izlaganje nastavnika i shvataju da editor Bloka pruža mogućnost kreiranja programa tako što prevlačimo odgovarajuće blokove iz palete blokova i povezujemo ih. Blokove povezujemo da bismo kreirali program koji će biti pokrenut na Micro:bit uređaju.

Učenici otvaraju kategorije nasumično i analiziraju blokove. Uočavaju da blokovi mogu biti samostalni blokovi koji odgovaraju naredbama jezika, ali i blokovi kojima su definisani događaji (dugme, pomjeranje...).

Aktivnost 4

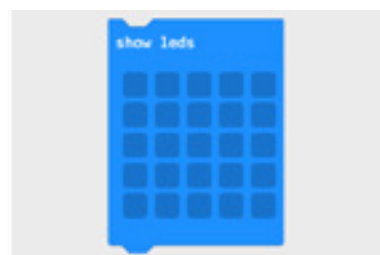
Kategorija Basic – Učenici otvaraju *Basic* kategoriju u kojoj se nalaze blokovi sa osnovnim funkcijama za rad sa Micro:bit-om i upoznaju blokove:



Prikazivanje numeričkih vrijednosti na ekranu Micro:bit uređaja.



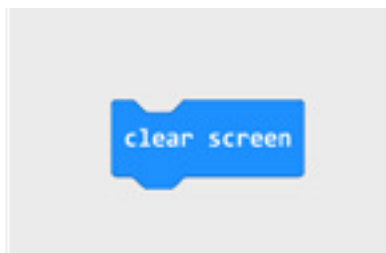
Prikazivanje ikona iz padajuće liste na ekranu Micro:bit uređaja.



Prikazivanje slike na ekranu Micro:bit uređaja.



Prikazivanje niza karaktera - teksta (stringa) na ekranu.



Brisanje (isključivanje) led dioda na ekranu.



Blok ponavljanja, program se izvršava dok se uređaj ne isključi.



Pokretanje programa prilikom uključivanja Micro:bit uređaja.



Definisanje pauze za vremenski period u milisekundama.



Prikazivanje strelica na ekranu Micro:bit uređaja.

Aktivnost 5

Zadatak – Nastavnik postavlja zadatak u vezi sa upotrebom osnovnih blokova u jednostavnom kôdu. Potrebno je da učenici kreiraju program kojim će se na ekranu Micro:bit uređaja prikazati poruka „Zdravo, svijete!“ i to pri uključivanju uređaja.

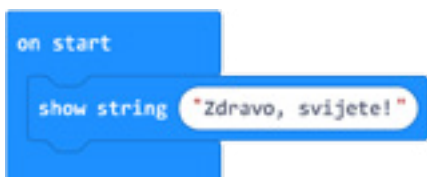
Učenici, uz instrukcije nastavnika, otvaraju kategoriju *Basic*, zatim lijevim klikom prenose na radnu površinu blok *on start*.




Ponovnim klikom na kategoriju *Basic* biraju blok *show string* i uklapaju ga u prvi. Riječ „Hello“ mijenjaju u „Zdravo, svijete!“



Izgled konačnog kôda:



Simulator – Nakon završenog zadatka, učenici testiraju kôd. Saznaju da postoje dvije mogućnosti za testiranje kôda.

Prva mogućnost je da program pokrenu u simulatoru, klikom na dugme  koje se nalazi ispod prozora simulatora.

Aktivnost 6

Programiranje Micro:bit uređaja – Učenici saznaju da je druga mogućnost da testiraju kôd da ga prebace na Micro:bit. Učenici priključuju Micro:bit na računar korišćenjem USB kabela. Klikom na dugme *download*, na računaru se preuzima .hex fajl. Zatim je potrebno sa računara prebaciti (kopirati ili premjestiti) .hex fajl na Micro:bit. Prilikom prebacivanja programa, na Micro:bit uređaju treperi žuta LED dioda, koja se nalazi sa zadnje strane, a koja označava da je transfer podataka (datoteke) u toku. Kada dioda prestane da treperi, to je znak da je program prebačen na uređaj i da ga možemo pokrenuti na istom.

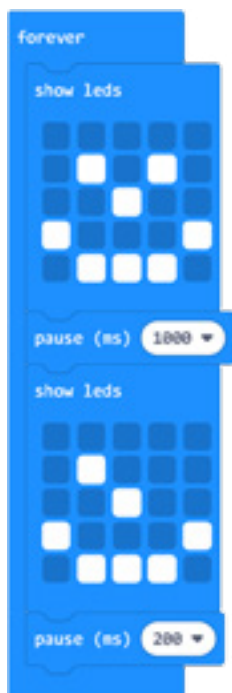


Učenici zaključuju da Micro:bit uređaj može da čuva samo jedan program. Uređaj će izvršavati određeni program sve dok se ne unese novi program.

Aktivnost 7

Dodatni zadatak – Učenici kreiraju program kojim će Micro:bit, svojim svijetlećim diodama, prikazivati smješka koji neprekidno namiguje i to tako što svake sekunde namigne lijevom okom, korišćenjem naredbi iz kategorije *Basic*:

- forever,
- show leds,
- pause (ms).



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_7xHKRJibyVsf



PLAN ČASA BROJ 3	
OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da razlikuje kategorije blokova u MakeCode okruženju i upotrebljava blokove iz kategorija <i>Input</i> i <i>Music</i> .
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, internet konekcija.
KRATAK OPIS ČASA	
Kroz niz aktivnosti učenici se upoznaju sa kategorijama blokova u Microsoft MakeCode okruženju. Upoznaju funkcije blokova i kreiraju kôd. Shvataju značenje pojmova <i>string</i> i <i>linijski program</i> .	
Napomena: Učenike upoznati sa blokovima koji se koriste u V razredu. Ostale blokove predstaviti po procjeni nastavnika.	

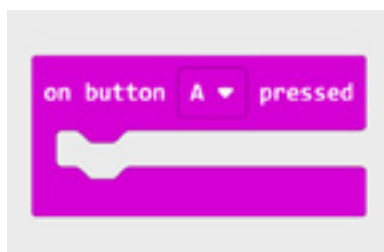
Aktivnost 1

Upoznavanje sa kategorijama blokova – Učenici se upoznaju sa kategorijama blokova u Microsoft MakeCode okruženju tako što prate rad nastavnika koji putem direktne demonstracije prikazuje i objašnjava funkcije blokova.

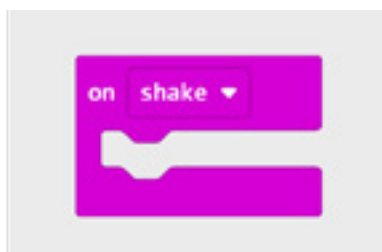
Učenici idu na internet adresu <http://www.makecode.com> i pokreću novi projekat.

Aktivnost 2

Kategorija Input – Učenici otvaraju *Input* kategoriju u kojoj se nalaze blokovi kojima se definišu događaji (klik na dugme, pokreti,...) i čuvaju podaci senzora. Prate izlaganje nastavnika i upoznaju blokove:



Izvršavanje koda kada je pritisnuto dugme A, B ili istovremeno A i B.



Izvršavanje koda kada se desi neki pokret.



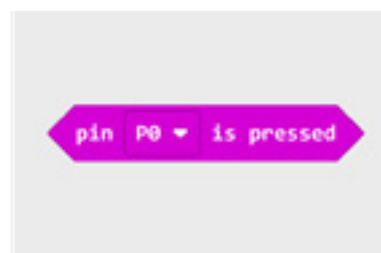
Pokreće obrađivač događaja (dio programa koji će se pokrenuti kada se nešto dogodi, na primjer kada se pritisne dugme).



Prihvata stanje dugmeta A i B, bilo da su pritisnuti ili ne.



Čuva vrijednost ubrzanja.



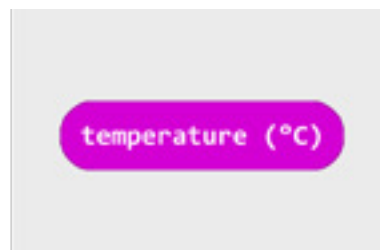
Prima vrijednost stanja pina.



Čuva nivo osvijetljenja u intervalu od 0 do 255.



Prihvata trenutne vrijednosti kompasa u stepenima.



Čuva vrijednost temperature u Celzijusima.



Testira da li je došlo do nekog pokreta.



Pokreće neki kod kada mikrofoni registruje zvuk.



Pokreće neki kod u programu kada se logo Micro:bit pritisne, dodirne ili pusti.



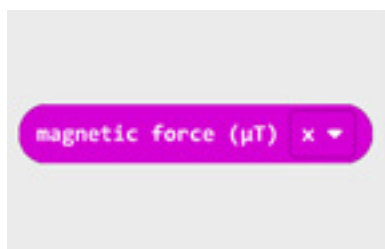
Provjerava da li je logo Micro:bit trenutno pritisnut.



Otkriva koji je nivo zvuka koji se čuje iz mikrofona.



Čuva vrijednost u stepenima kada uređaj rotira oko x i y ose.



Čuva vrijednost magnetne sile.



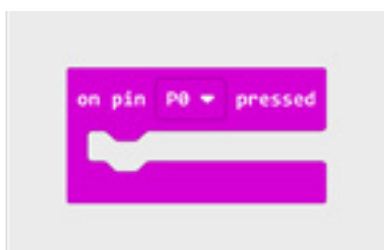
Čuva broj milisekundi od trenutka kada je uređaj uključen.



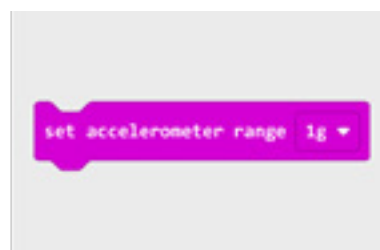
Čuva broj mikrosekunde kada je uređaj uključen.



Pokreće sekvencu kalibracije kompasa.



Izvršavanje koda kada je pritisnut neki od pinova.



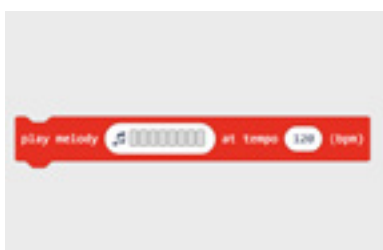
Podešava dio uređaja koji mjeri ubrzanje.



Govori koliko glasno treba da bude da vaša ploča otkrije glasan zvuk.

Aktivnost 3

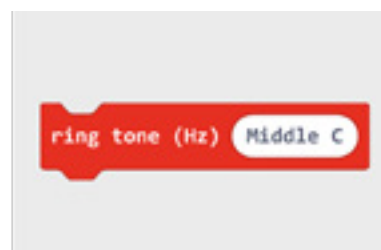
Kategorija Music – Klikom na kategoriju *Music*, učenici upoznaju blokove za rad sa muzikom.



Svira kratku melodiju od nota komponovanih u nizu.



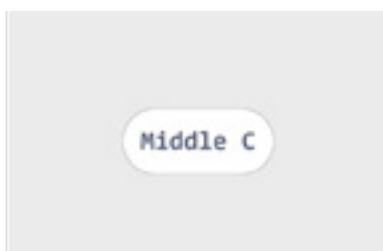
Pušta ton određene dužine preko pina P0.



Pušta ton preko pina P0.



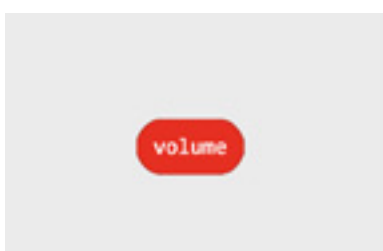
Pauza za određeni vremenski period preko pina P0.



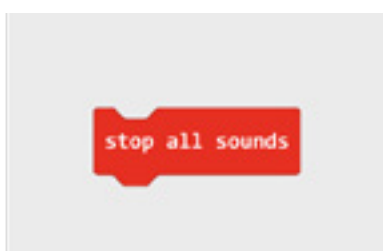
Preuzima frekvenciju muzičke note.



Podešava jačinu zvuka za sintisajzer zvuka.



Preuzima trenutni nivo jačine zvuka sintisajzera zvuka.



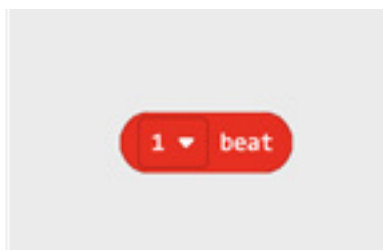
Zaustavlja sve zvukove koji se trenutno reprodukuju i sve druge koji čekaju da se reprodukuju.



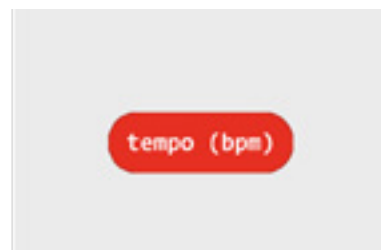
Promjena tempa za određenu vrijednost.



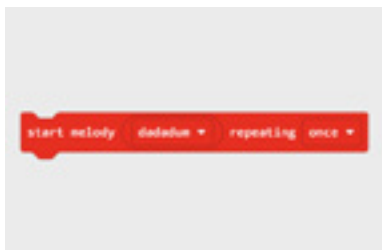
Postavlja tempo na određenu vrijednost.



Vraća vrijednost trajanja bita u milisekundama.



Vraća vrijednost tempa po bitima u sekundi.



Započinje sviranje melodije.



Zaustavlja sviranje melodije.



Podiže događaje za melodije ili muzičke događaje.



Reprodukuje zvučni zapis dok se ne završi.



Reprodukuje zvučni izraz.



Omogućava zvučniku na uređaju da reprodukuje muziku i zvukove.

Aktivnost 4

String – Učenici usvajaju pojam „string“, osvrćući se na jednostavan program koji su kreirali na prethodnom času – program za prikazivanje poruke „Zdravo, svijete!“. Znaju da su tu koristili jednu vrstu podataka pod navodnicima, koji se u programskim jezicima nazivaju string. Shvataju da string predstavlja niz karaktera, predstavljenih između znakova navoda. Stringovi imaju veliku primjenu u programiranju. Za Micro:bit uređaj podržani su samo ASCII kodovi znakova od 32 do 126. U te znake spadaju slova engleske abecede, cifre, interpunkcijski znaci i nekoliko simbola. Svi ostali znaci pojavljuju se kao ? na ekranu Micro:bit-a.

Linijski program – Nastavnik prikazuje nekoliko kodova u kojima su blokovi poređani jedan za drugim. Učenici uočavaju da se blokovi izvršavaju po redoslijedu po kome su navedeni i zato se nazivaju linijski programi.

Aktivnost 5

Zadatak – Nastavnik postavlja zadatak učenicima. Potrebno je da se sjete pjesme koju su učili iz muzičke kulture „Na kraj sela žuta kuća“ i napišu program za Micro:bit uređaj kojim će napraviti melodiju pjesme.

Da bi se prisjetili melodije, pjesmu će naći na adresi https://www.youtube.com/watch?v=dpXh_jxoldc

Melodija pjesme – Učenici razmišljaju o problemu i analiziraju korake kojima će doći do rješenja. Zaključuju da moraju znati notni zapis pjesme da bi napisali melodiju.

Posmatraju notni zapis i zapisuju note:

c, d, e, e
c, d, e, e
c, d, e
c, d, e, e
c, d, e
e, e, d, d
c



Učenici pokreću novi projekat klikom na dugme *New Project* i pišu naziv *Žuta kuća*. Znaju da moraju definisati ulaz, pa u tu svrhu koriste iz grupe osnovnih blok *on start*. Kao što znaju, da bismo odsvirali neki ton ili melodiju, moramo upotrijebiti blokove iz kategorije *Music*. Pomoću bloka pjesma reprodukuje ton određene visine i trajanja. Prva nota koju prevlačimo u blok *on start* biće nota C, dužine jednog bita (otkucaja), ili tačnije jedne sekunde.

Sada je potrebno samo prevući ispod datog bloka ostale note C, D, E, E.



Kompletan program – Učenici na isti način ređaju blokove za ostale note i dobijaju kôd na slici lijevo.

Na simulatoru slušaju melodiju i dolaze do zaključka da kôd ne reprodukuje tačnu melodiju. Upoređuju kôd sa notnim zapisom i otkrivaju grešku u programu.

Greška – neki blokovi traju kraće od potrebnog. Ispravljaju greške – promijeniće trajanje nota i uskladiti ih sa notnim zapisom, po principu 1 otkucaj = 1 beat. Izgled izmijenjenog kôda – slika desno.



Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_ixHc4rUxgXt5



Aktivnost 6

Postupak rješavanja bilo kog zadatka (problema) – Učenici su se kroz prethodni zadatak upoznali sa pojmom „greška” (ili „buba”, engl. *bug*), kao i načinom otklanjanja grešaka ili „debugovanja računarskog programa”. Došli su do zaključka da postupak rješavanja bilo kog problema podrazumijeva prelazak kroz svaku od sljedećih faza:

Faza 1: Razmisli o problemu i napiši ili nacrtaj algoritam (korake) za njegovo rješavanje.

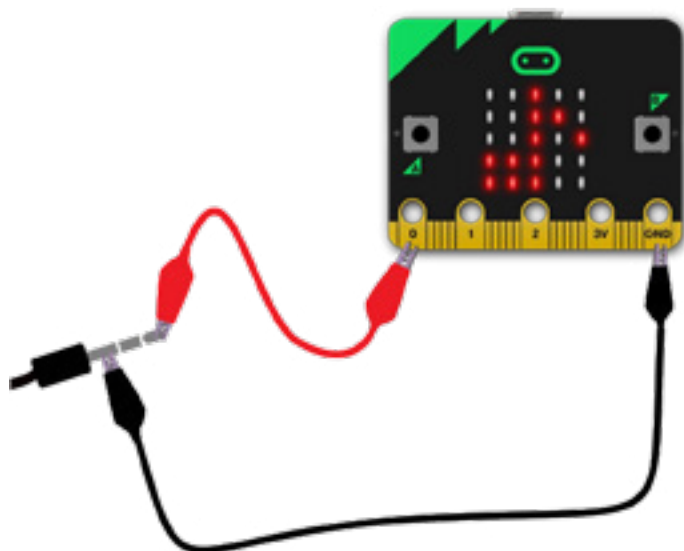
Faza 2: U skladu sa algoritmom, odaberi i složi blokove.

Faza 3: Testiraj i analiziraj rad programa. Ispravi greške (ako ih ima).

Aktivnost 7

Upotreba dodatne opreme (za one koji žele više) – Nakon programiranja Micro:bit uređaja, učenici potvrđuju ranije stečena znanja, konkretno u ovom slučaju uočavaju da Micro:bit nema zvučnik.

Zaključuju da im je potreban zvučnik ili slušalice i krokodil-kablovi, kojima će ih povezati na Micro:bit uređaj. Način vezivanja zvučnika ili slušalica na Micro:bit uređaj prikazan je na slici.

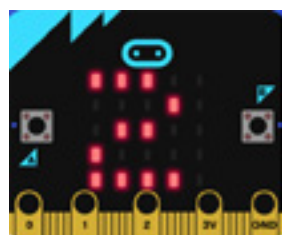






Donji dio priključka (najbliži plastici) vezujemo na GND pin, a vrh priključka na pin P0.

Test za učenike:

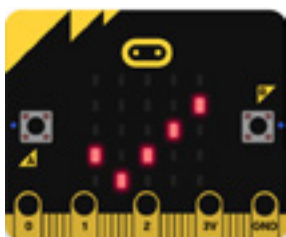
Provjeri koliko znaš – zaokruži tačne odgovore:

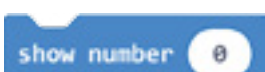



1. Koji blok treba da upotrijebiš da bi na ekranu bio prikazan broj 2?




- A. 
- B. 
- C. 
- D. 

2. Koji blok treba da upotrijebiš da bi na ekranu bio prikazan „check“ znak kao na slici?



- A. 
- B. 
- C. 
- D. 

3. Kojim blokom utvrđujemo da li je na Micro:bit uređaju pritisnuto dugme A, B ili oba istovremeno?

- A. 
- B. 
- C. 

4. Šta će se desiti kad Micro:bit programiramo prikazanim kodom?



- A. Biće odsvirana nota C dužine 1.000 milisekundi
- B. Biće odsvirana nota C dužine 10 milisekundi
- C. Biće odsvirana nota C dužine 10 sekundi
- D. Neće se desiti ništa

5. Koji blok treba da upotrijebiš za puštanje tona ograničenog trajanja?

- A. 
- B. 
- C. 

TAČNI ODGOVORI

1 – D	2 – C	3 – C	4 – A	5 – A
-------	-------	-------	-------	-------

PLAN ČASA BROJ 4

OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da napiše program koristeći blokove iz kategorija <i>Basic</i> i <i>Input</i> .
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, internet konekcija.

KRATAK OPIS ČASA

Učenici, uz pomoć nastavnika, pišu programe koristeći blokove iz dvije kategorije. Kroz zadatke učenici će proširiti znanja o ulaznim i izlaznim uređajima.

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik kroz zadavanje zadataka i praćenje načina rješavanja koordinira rad učenika. Učenici pokreću novi projekat kojem dodjeljuju naziv „Vježbanje”.

Zadatak 1. – Učenici dobijaju zadatak da programiraju Micro:bit tako da se prilikom uključivanja uređaja na ekranu pojavi poruka „Zdravo” na engleskom jeziku – „Hi”.

Učenici otvaraju kategoriju *Basic*, zatim lijevim klikom prevuku blok *on start* i postave ga na radnu površinu. Na isti način, prevlačenjem, u blok *on start* ugrade blok *show string* i napišu riječ *Hi*.



Aktivnost 2

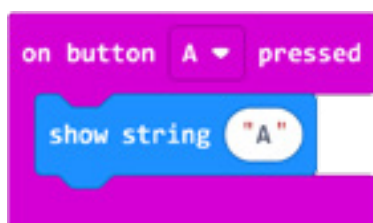
Zadatak 2. – Drugi kôd koji učenici treba da prikažu je srce koje neprestano stoji na ekranu. Učenici otvaraju kategoriju *Basic*, zatim lijevim klikom prevuku blok *forever* i postave ga na radnu površinu. Na isti način, prevlačenjem, u blok *forever* ugrade blok *show icon* i iz padajućeg menija izaberu (ostave) ikonu srca.



Aktivnost 3

Zadatak 3. – Učenici imaju zadatak da napišu program kojim će se, pritiskom na dugme A, na ekranu pojaviti slovo A.

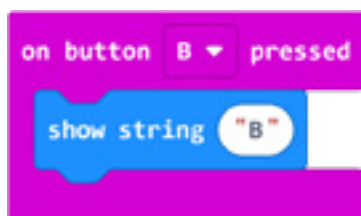
Učenici otvaraju kategoriju *Input*, zatim lijevim klikom prevuku blok *on button A pressed* i postave ga na radnu površinu. Novim prevlačenjem, u blok *on button A pressed* ugradiće blok *show string* i napisati slovo A.



Aktivnost 4

Zadatak 4. – Učenici imaju zadatak da napišu program kojim će se, pritiskom na dugme B, na ekranu pojaviti slovo B.

Učenici otvaraju kategoriju *Input*, zatim lijevim klikom prevuku blok *on button A pressed* i postave ga na radnu površinu. Iz padajućeg menija umjesto slova A izaberu slovo B. Novim prevlačenjem, u blok *on button B pressed* ugradiće blok *show string* i napisati slovo B.



Aktivnost 5

Zadatak 5. – Učenici imaju zadatak da napišu program kojim će se istovremenim pritiskom na dugmad A i B na ekranu pojaviti ikona note.

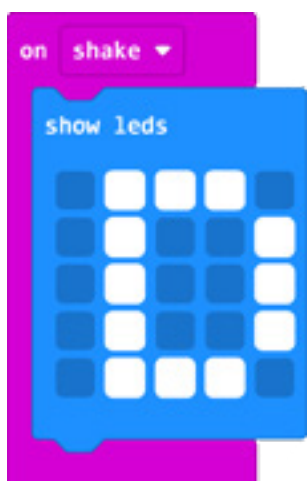
Iz kategorije *Input* učenici biraju blok *on button A pressed* i postavljaju ga na radnu površinu. Iz padajućeg menija umjesto slova A izaberu opciju A+B. Novim prevlačenjem, u blok *on button A+B pressed* ugradiće blok *show icon* i iz padajućeg menija izabрати ikonu note.



Aktivnost 6

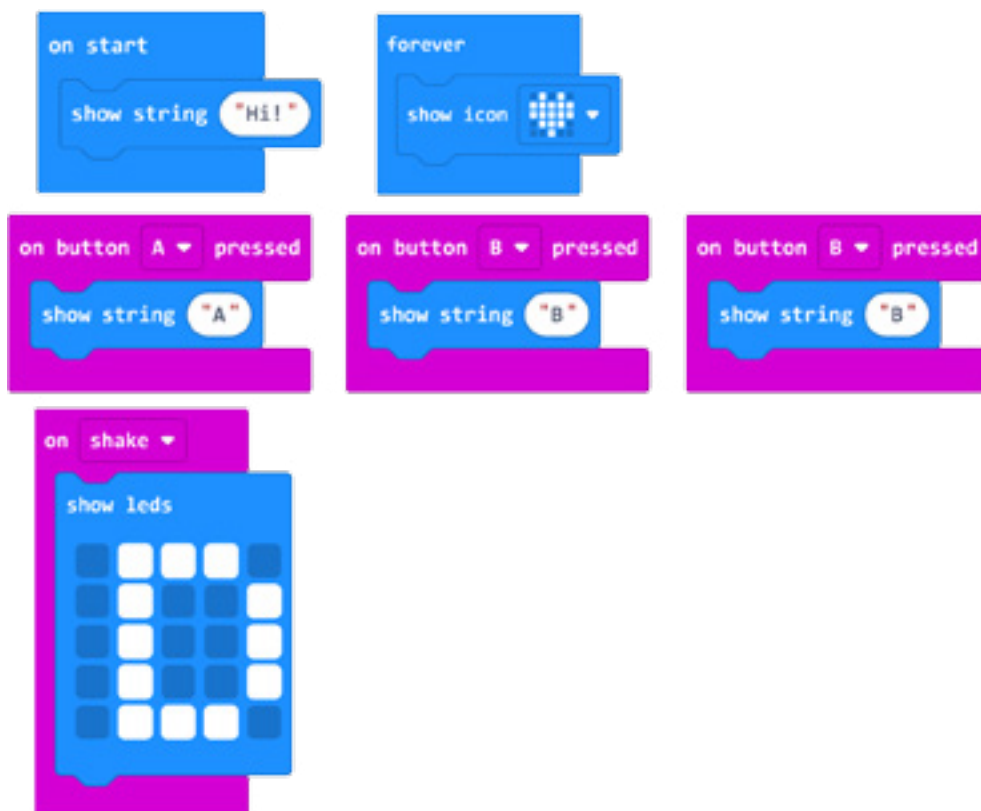
Zadatak 6. – Učenici imaju zadatak da napišu program kojim će se, kada protresu Micro:bit uređaj, na ekranu pojaviti početno slovo njihovog imena, koje će sami ispisati paleći odgovarajuće diode.

Iz kategorije *Input* učenici biraju blok *on shake* i postavljaju ga na radnu površinu. Novim prevlačenjem, u blok *on shake* ugradiće blok *show leds* i uključiti LED diode tako da se na ekranu prikaže prvo slovo njihovog imena.



Aktivnost 7

Testiranje i analiza programa – Učenici prvo testiraju program pomoću simulatora, a zatim programiraju Micro:bit uređaj. Kako bi program prebacili na Micro:bit, treba da ga prikaže na računar korišćenjem USB kabla. Klikom na dugme *download* preuzimaju .hex fajl na njihov računar, a zatim na Micro:bit uređaj. Program se može direktno preuzeti/sačuvati na uređaju. Učenici testiraju program na uređaju, korak po korak.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_TpKE1yA5mgyu



Aktivnost 8

Dodatni zadatak – Učenici rade samostalno i provjeravaju tačnost.

1. Ovim kôdom će biti prikazan broj 7 i ikonica makaze.



Ispravi kôd tako da se ikonica makaze prikazuje 3 sekunde nakon prikazivanja broja 7. Uporedi svoje rješenje sa ovim koje se nalazi na linku: https://makecode.microbit.org/_8Rc9rfA7wHC6

ili otvori rješenje čitanjem QR kôda.



PLAN ČASA BROJ 5	
OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da programira Micro:bit koristeći novi blok iz kategorije <i>Variables</i> .
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, internet konekcija, nešto čime ćete pričvrstiti Micro:bit za cipelu ili nogu – žica, ljepljiva ili čičak traka.
KRATAK OPIS ČASA	
Učenici će shvatiti kako računari koriste promjenljive za čuvanje i pozivanje informacija koje se mijenjaju. U ovom slučaju, riječ je o broju koraka koje smo napravili. Znaće kako da kreiraju i koriste smislene nazive za promjenljive, npr. „koraci”, da bi pratili brojanje koraka.	
Napomena: Aktivnosti koje su predložene u Planu časa broj 5 mogu se realizovati po procjeni nastavnika sa naprednim učenicima, s obzirom na to da se koriste promjenljive (po planu predviđene za VI razred). Primjenjujući princip individualizacije nastave, ponavljati zadatke iz prethodnih časova.	

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. U skladu sa horizontalnom korelacijom, koristeći stečena znanja iz matematike, uvodi se pojam promjenljiva. Učenici su iz matematike kroz rješavanje nejednačina upoznati sa pojmom promjenljiva, pa se nastavniku preporučuje da se zadrži na tom nivou, uz napomenu da će više o ovome učiti u starijim razredima. Učenici zaključuju da se vrijednost dodjeljuje promjenljivoj (u ovom slučaju je zadana na 0) kada se program prvi put pokrene. Dolaze do zaključka da su promjenljive objekti koji čuvaju neku vrijednost, koja se može mijenjati tokom izvršavanja programa.

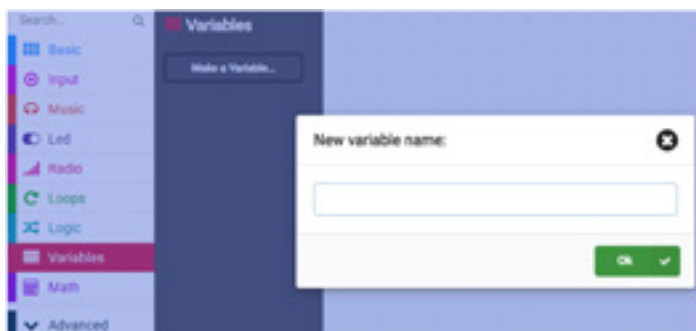
Učenici pišu program takav da, kada se protrese Micro:bit (napravi se jedan korak), promjenljiva koja broji korake poveća se za 1; na ekranu se neprekidno prikazuje trenutna vrijednost promjenljive, tj. trenutni broj učinjenih koraka.

Aktivnost 2

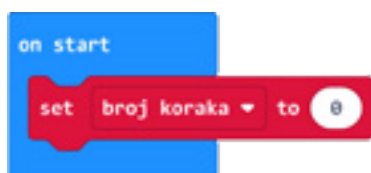
Prvi korak u pisanju kôda – Učenici će iz kategorije *Basic* izabrati *on start* naredbu. Znaju da će se sve naredbe koje stave u ovaj blok pokrenuti kada se pokrene Micro:bit.



Drugi korak u pisanju kôda – Učenici će prvo kreirati promjenljivu u kojoj će čuvati broj koraka. U kategoriji *Variables* kliknuće na *Make a variable*, u ponuđenu kućicu upisaće ime nove promjenljive „broj koraka”, a zatim će kliknuti na OK. Nakon toga, ponovnim klikom na kategoriju *Variables*, kreirani blok *broj koraka* će biti u ponudi.

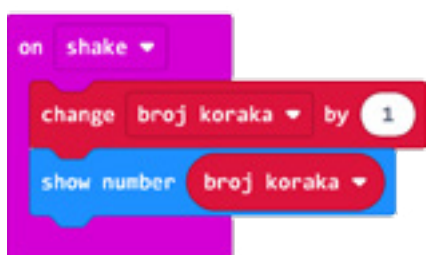


Treći korak u pisanju kôda – Učenici će u kategoriji *Variables* izabrati i kliknuti na naredbu *set broj koraka to 0* i ugradiće je u blok *on start*.



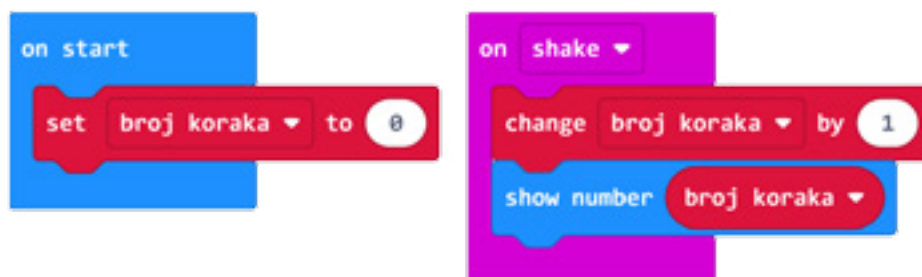
Četvrti korak u pisanju kôda – Sljedeći zadatak učenicima je da nastave kôd tako da se promjenljiva „broj koraka” povećava za 1 kad se Micro:bit protrese (svakim učinjenim korakom).

Iz kategorije *Input* izabraće naredbu *on shake* u koju će ugraditi dva bloka. Prvi blok je *change 'broj koraka' by 1* iz kategorije *Variables*, a drugi blok je *show number* iz kategorije *Basic*, u koji je ugrađena promjenljiva „broj koraka”.



Aktivnost 3

Testiranje programa – Učenici prvo testiraju program pomoću simulatora, a zatim programiraju Micro:bit uređaj, koji će žicom, ljepljivom ili čičak trakom pričvrstiti za nogu. Učenici testiraju program na uređaju.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_10z4fk68Y1VX



Aktivnost 4

Zadatak – Učenici će programirati Micro:bit da vrši funkciju štoperice, tako što će se tajmer pokrenuti pri uključivanju Micro:bit-a, a vrijednost promjenljive u sekundama (proteklo vrijeme) prikazivaće se neprekidno. Prvo će kreirati novu promjenljivu koju mogu nazvati „vrijeme”.



Promjenljivoj „vrijeme” dodijeliće vrijednost koju žele da je na početku. S obzirom na to da prave štopericu, vrijednost promjenljive će biti 0.

Koristiće blok *set vrijeme to*, koji će staviti u *on start* blok. Na ovaj način dodijeliće promjenljivoj početnu vrijednost koju će dobiti čim se Micro:bit uključi.



Učenici zaključuju da im je za praćenje podataka neophodno da se na ekranu cijelo vrijeme ispisuje vrijednost promjenljive. Za to će koristiti blok *show number*. U polje za upisivanje broja privući će blok iz grupe *Variables*, koji nosi vrijednost njihove promjenljive. Dobili su kôd:



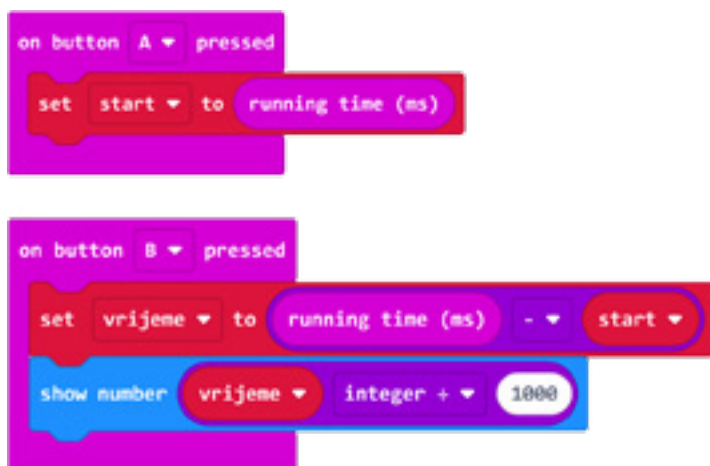
Učenici analiziraju problem i zaključuju da je potrebno da se broj koji se ispisuje na ekranu povećava za jedan nakon svake sekunde da bi štoperica bolje radila. To će im omogućiti promjenljiva „vrijeme”, koju su već napravili. Kada se na ekranu ispiše prvi broj, u ovom slučaju 0, napraviće pauzu od jedne sekunde. Nakon toga povećaće vrijednost promjenljive „vrijeme” za 1. To će uraditi pomoću bloka *change vrijeme by*, u koji upisuju 1.

S obzirom na to da su ovi blokovi smješteni unutar bloka *forever*, ova štoperica će raditi dok ne isključe Micro:bit.



Aktivnost 5

Dodatni zadatak za one koji žele više – Napiši kôd kojim ćeš Micro:bit programirati da vrši funkciju štoperice, tako što će odbrojavanje vremena početi klikom na dugme A, a proteklo vrijeme prikazaće se klikom na dugme B.



Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_LDf3um5wxVMm



PLAN ČASA BROJ 6	
OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da napiše program koristeći blokove iz više kategorija.
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, internet konekcija.
KRATAK OPIS ČASA	
Učenici, uz pomoć nastavnika, pišu programe koristeći blokove iz dvije ili više kategorija. Kroz zadatke učenici će proširiti znanja o ulaznim i izlaznim uređajima, kao i o nekim senzorima.	

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa zahtjevima zadataka. Potrebno je kreirati program kojim ćemo iskoristiti jednu od funkcija Micro:bit-a kao ulaznog uređaja. Tačnije, Micro:bit vrši određene radnje na osnovu informacija koje dobija od korisnika ili okoline (način startovanja programa). Analizirajući blokove, učenici uočavaju da kategorija *Input* sadrži naredbe, koje na osnovu informacija koje Micro:bit uređaj dobija od korisnika ili okoline putem pritiska na dugme A, B ili A+B, kao i senzora za svjetlost, temperaturu, akcelerometar, pokreću određeni program.

Aktivnost 2

Zadatak – Učenici dobijaju zadatak da programiraju Micro:bit tako da pritiskom na dugme A prikažu nivo osvjetljenja u prostoriji, a da se pritiskom na dugme B prikazuje temperatura u celzijusima.

Nakon razmatranja problema, učenici definišu korake kojima će doći do rješenja:

Korak 1: Pritisni taster A.

Korak 2: Prikaži trenutni nivo osvjetljenja u prostoriji.

Korak 3: Pritisni taster B.

Korak 4: Prikaži trenutnu temperaturu u prostoriji.

Kôd za mjerenje nivoa osvjetljenja – Na internet stranici koju su učenici otvorili, na adresi <https://makecode.microbit.org>, pokreću novi projekat klikom na dugme *New Project* i upisuju ime projekta *Svjetlost i temperatura*.

Učenici, uz instrukcije nastavnika, pišu kôd, uz analizu svakog bloka koji upotrijebe. Da bismo omogućili da Micro:bit primi informaciju od korisnika, pritiskom korisnika na taster A, iz kategorije *Input* biramo i prevlačimo u prostor za programiranje blok:



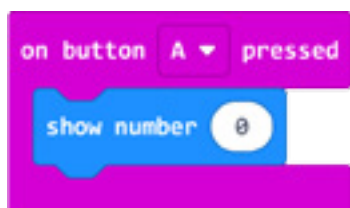
Iz padajuće liste biramo taster A.

Da bismo prikazali nivo osvjetljenja u prostoriji, prvo treba da u postavljeni blok prevučemo blok iz kategorije *Basic*, koji se koristi za prikazivanje numeričkih vrijednosti.



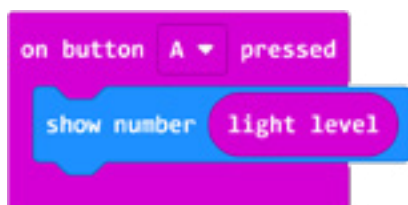
Koristimo ovaj blok za prikaz jer je trenutni nivo osvjetljenja u prostoriji predstavljen kao numerička vrijednost od 0 do 255, gdje 0 označava da je mrak u prostoriji, a 255 označava jako osvjetljenje.

Izgled kôda:



Vrijednost nivoa osvjetljenja u prostoriji „čuva“ se u bloku *light level*, određenom za numeričku vrijednost. Učenici uočavaju da blok čuva očitane vrijednosti senzora za svjetlost, koji je u stvari ekran Micro:bit uređaja. Znaju da dio LED dioda ekrana služi i kao senzor za svjetlost.

Konačan izgled kôda:



Aktivnost 3

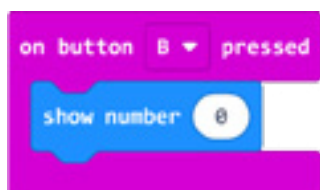
Kôd za mjerenje temperature – Učenici, uz instrukcije nastavnika, pišu kôd – pritiskom korisnika na taster A, iz kategorije *Input* biramo i prevlačimo u prostor za programiranje bloka *on button A pressed*. Iz padajuće liste biramo taster B.



Da bismo prikazali temperaturu u prostoriji, prvo treba da u gornji blok prevučemo blok iz kategorije *Basic*, koji se koristi za prikazivanje numeričkih vrijednosti.



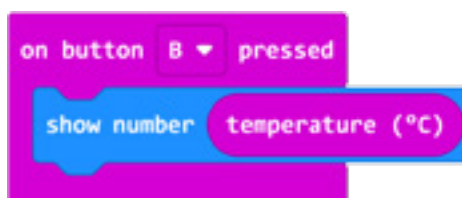
Izgled kôda:



Vrijednost trenutne temperature u prostoriji „čuva“ se u bloku *shownumber*, određenom za numeričku vrijednost.

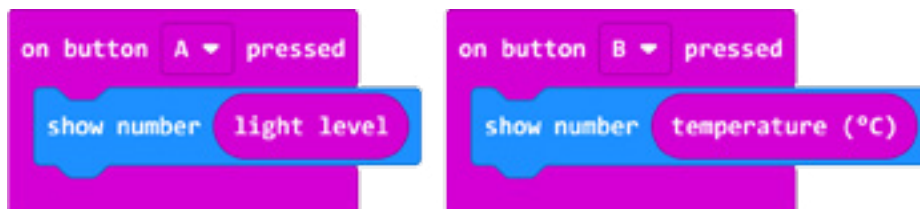
Blok *temperature (°C)* čuva očitanu vrijednost senzora za temperaturu Micro:bit-ovog procesora, jer ne postoji poseban senzor za temperaturu koji je ugrađen u Micro:bit uređaj. Vrijednost očitana sa senzora prikazuje se u celzijusima.

Izgled gotovog dijela kôda:

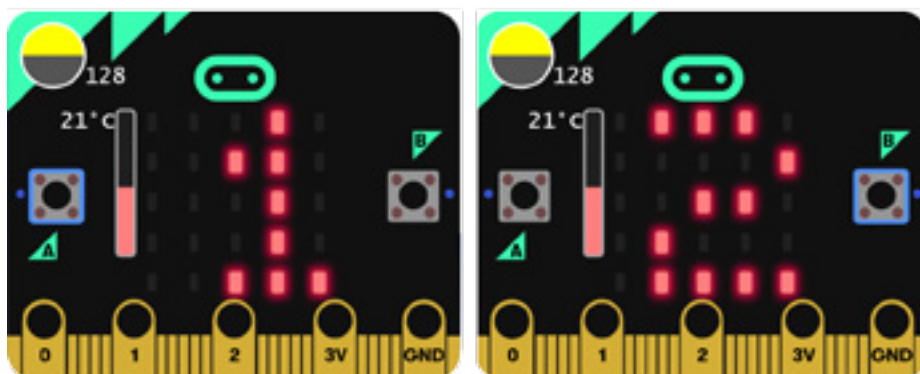


Aktivnost 4

Testiranje i analiza programa –



Učenici prvo testiraju program pomoću simulatora, a zatim programiraju Micro:bit uređaj. Kako bi program prebacili na Micro:bit, treba da ga prikače na računar korišćenjem USB kablova. Klikom na dugme *download*, preuzmaju .hex fajl na njihov računar. Prevlačenjem fajla na Micro:bit, uređaj je spreman za rad. Učenici testiraju program na drugi način.



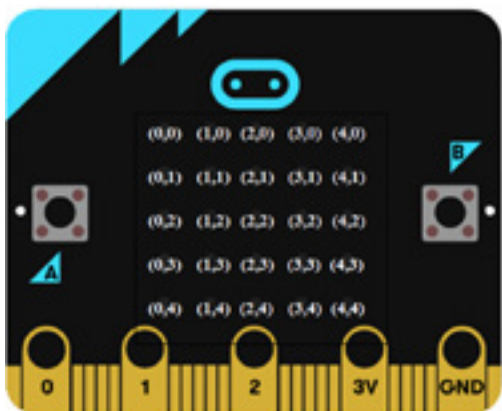
Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_dPU3oRd565t7



PLAN ČASA BROJ 7	
OVI 6	Na kraju učenja učenik će biti osposobljen da razumije koncepte programa i ulaznih i izlaznih uređaja.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da radi sa blokovima za kontrolu LED dioda u koordinatnom sistemu Micro:bit-a.
Nastavna sredstva i pomagala	Računar, projektor, Micro:bit uređaj, internet konekcija.
KRATAK OPIS ČASA	
<p>Učenici se upoznaju sa položajem LED dioda u koordinatnom sistemu. Shvataju funkcije nekih blokova iz kategorija <i>Led</i> i <i>Radio</i> i koriste ih za programiranje Micro:bit-a.</p> <p>Učenici ne znaju šta je koordinatni sistem, pa je poželjno pojam koordinata učenicima objasniti na drugačiji način. Jedan od načina: da bi razlikovali LED diode – svaka od njih ima svoje ime, a to su dva broja. Tako imamo ime za diodu (3,2) i slično. Imena svih dioda su prikazana na slici. Umjesto „ime diode“ koristićemo i termin „koordinate diode“.</p> <p>Napomena: Učenike upoznati sa blokovima koji se koriste u V razredu. Ostale blokove predstaviti po procjeni nastavnika.</p>	

Aktivnost 1

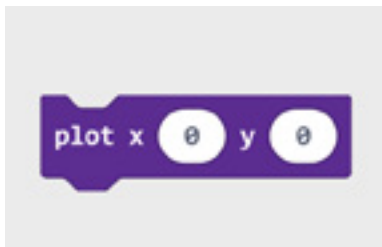
Upoznavanje sa LED diodama – Učenici se upoznaju sa položajem LED dioda u koordinatnom sistemu tako što analiziraju sliku sa koordinatama:



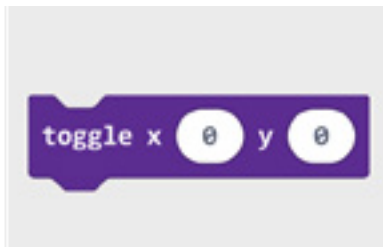
Uočavaju da su LED diode raspoređene u mreži 5 x 5. Učenici su ranije naučili da se svaka LED dioda može pojedinačno uključiti i isključiti. Tako mogu da „crtaju“ znakove u obliku srca ili osmijeha. Uočavaju i da se koordinatni početak (0,0) nalazi u gornjem lijevom uglu mreže LED dioda. Vrijednost x koordinata (tj. prvog broja) kreće se od 0 do 4 i povećava se slijeva nadesno, dok se vrijednost y koordinata (tj. drugog broja) kreće od 0 do 4 i povećava se od vrha do dna.

Aktivnost 2

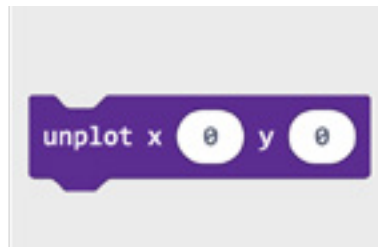
Kategorija Led – Klikom na kategoriju *Led*, učenici, uz demonstraciju nastavnika, upoznaju blokove za kontrolu rada LED dioda na ekranu Micro:bit-a. Učenici prate prezentovanje nastavnika.



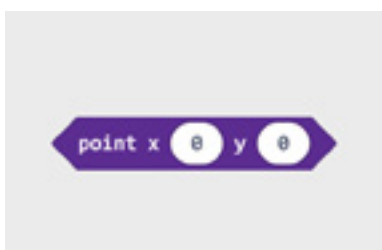
Uključuje LED diodu na koordinatama x i y.



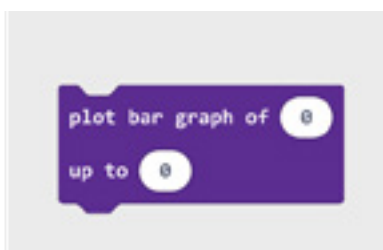
Mijenja aktuelno stanje diode (iz on u off i obrnuto)



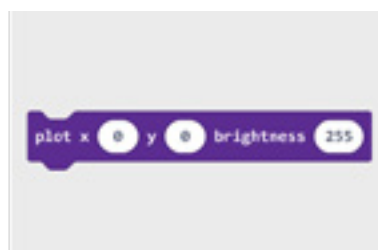
Isključuje LED diodu na koordinatama x i y.



Prikazuje stanje dioda (uključeno/ isključeno) na koordinatama x i y.



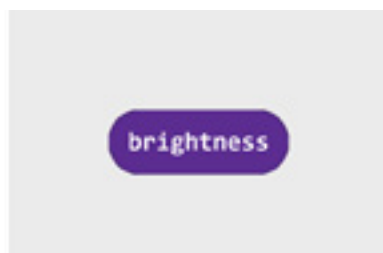
Prikazuje trakasti grafikon za vrijednost broja.



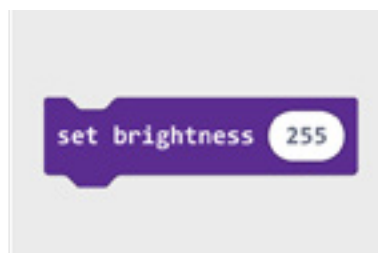
Uključuje LED diodu specifičnog intenziteta svjetlosti na koordinatama x i y.



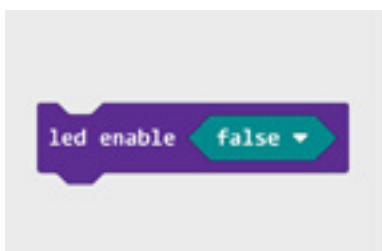
Uključuje LED diodu sa određenom osvjetljenošću na LED ekranu.



Pronalazi intenzitet LED ekrana kad je uključen.



Postavljanje intenziteta svjetlosti od 0 do 255.



Uključuje/isključuje LED ekran.



Zaustavlja animaciju koja se reprodukuje i sve animacije koje čekaju da se reprodukuju.



Podešava režim prikaza na crno-bijelo ili u sivim tonovima za prikazivanje LED dioda.

Aktivnost 3

Kategorija Radio – Klikom na kategoriju *Radio*, učenici otvaraju blokove za uspostavljanje radio-komunikacije između više Micro:bit uređaja. Tačnije, vrši se slanje i primanje podataka, uz korišćenje radio-paketa. Učenici prate prezentovanje nastavnika i upoznaju funkcije ovih blokova.



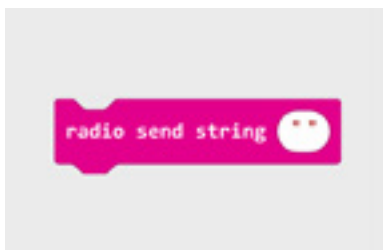
Postavljanje ID grupe za komunikaciju.



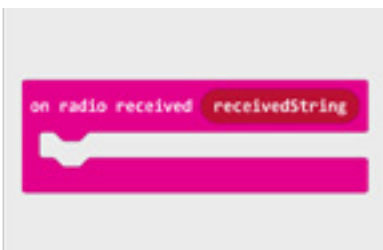
Šalje numeričku vrijednost preko radija svakom povezanom uređaju u grupi.



Šalje niz i broj zajedno putem radija drugim uređajima. Maksimalna dužina stringa je 8 znakova.



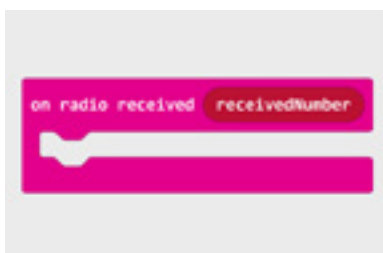
Šalje string preko radija svakom povezanom uređaju u grupi.



Registruje kod koji će biti pokrenut kada se primi string.



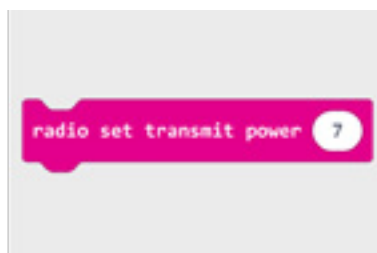
Pokreće dio programa kada Micro:bit primi par-ime-vrijednost preko radija.



Registruje kod koji će biti pokrenut kada se primi broj.



Preuzima jedno od svojstava iz posljednjeg primljenog radio paketa.



Čini radio signal uređaja jačim ili slabijim.



Čini da radio paket ugrađuje serijski broj ploče sa svakim paketom podataka.



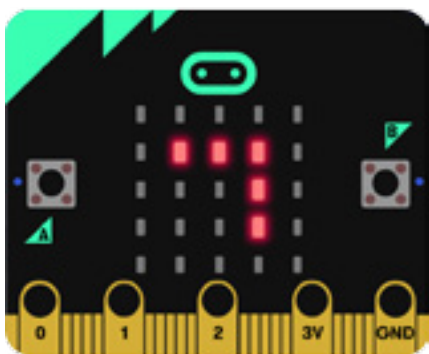
Mijenja opseg prenosa i prijema radija na dati kanal. Podrazumijevano je 7.



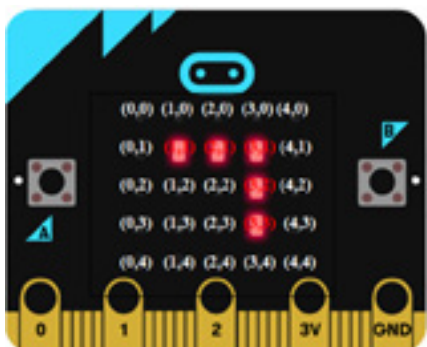
Šalje događaj preko radija da se podigne u sabirnici događaja drugog uređaja.

Aktivnost 4

Zadatak, rad sa LED diodama – Učenici će napisati program u kome će koristiti blokove za uključivanje/isključivanje LED dioda, pomoću koga će biti prikazana slika:



Razumijevanje problema – Učenici će za prikazivanje gornje slike koristiti (x,y) koordinate da bi odredili raspored LED dioda u mreži LED dioda. Koordinata x predstavlja horizontalni položaj (čije su vrijednosti 0, 1, 2, 3, 4), a koordinata y predstavlja vertikalni položaj (0, 1, 2, 3, 4). Prvo će na osnovu gornje slike definisati koordinate LED dioda koje su uključene.



Na osnovu slike učenici zaključuju da se uključene LED diode po horizontali nalaze na koordinatama (1,1), (2,1), (3,1), a uključene LED diode po vertikali na koordinatama (3,1), (3,2) i (3,3).


Pokretanje MakeCoda – Učenici će odabrati i slagati blokove u prostor za programiranje, uz instrukcije nastavnika. Da uključe LED diodu, koristiće blok.



Učenici su uključili označene LED diode i dobili konačan kôd:



Testiranje i analiza programa – Učenici testiraju program na dva načina:

1. pokreću program u simulatoru klikom na dugme  i
2. prebacuju program na Micro:bit, koji će nakon pokretanja pokazati da su uključene, tj. da svijetle diode kao na slici.

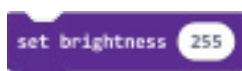


Aktivnost 5

Zadatak – Učenici kreiraju program kojim će podesiti intenzitet osvjetljenja uključene LED diode. Potrebno je da uključe LED diodu koja se nalazi na položaju (2,4) čiji je intenzitet osvjetljenja postavljen na 200.

Razumijevanje problema – Učenici analizom zadatka zaključuju da koordinata x predstavlja horizontalni položaj (čije su vrijednosti 0, 1, 2, 3, 4), a koordinata y predstavlja vertikalni položaj (0, 1, 2, 3, 4). Vrijednost nivoa osvjetljenja kreće se od 0 (crno) do 255 (svjetlo).

Pisanje kôda – Učenici pokreću novi projekat i postavljaju nivo osvjetljenja LED dioda koristeći blok *set brightness* iz kategorije *Led*, koji ugrađuju u blok *start*.



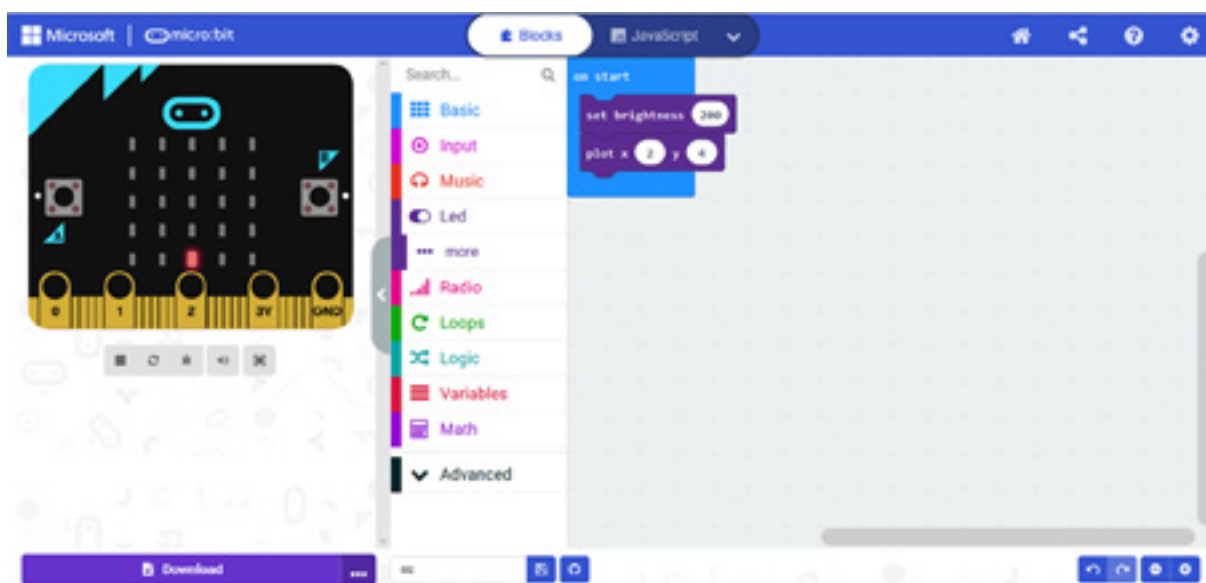
Postavljaju/upisuju nivo osvjetljenja 200.



Ispod nivoa osvjetljenja postavljaju blok i upisuju koordinate (2,4).



Izgled konačnog kôda:



Aktivnost 6

Dodatni zadatak za one koji žele više – Učenici će kreirati animaciju koja prikazuje „loading“ proces u obliku kružnice, koristeći blokove koji određuju koordinate LED dioda. Potrebno je da se animacija neprekidno ponavlja.

Prijedlog – koristiti blokove/naredbe: *forever*, *plot xy*, *pause (ms)*, *clear screen*.

Gotov kôd:



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_DhdF95ccFb63



Test za učenike:

Provjeri koliko znaš – zaokruži tačne odgovore:

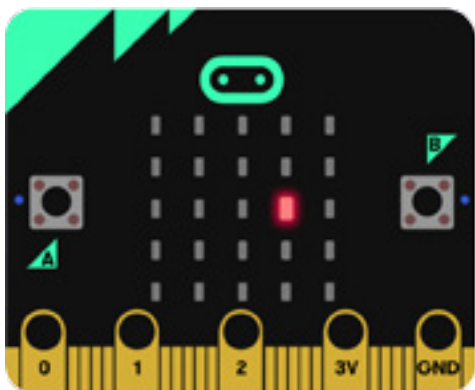
1. Koji je položaj (koordinate) gornje lijeve LED diode na ekranu Micro:bit-a?

- A. (1,1)
- B. (0,1)
- C. (0,0)
- D. (1,0)

2. Koji je položaj (koordinate) donje desne LED diode na ekranu Micro:bit-a?

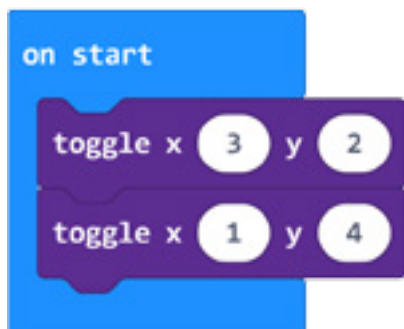
- A. (3,3)
- B. (4,4)
- C. (0,0)
- D. (1,1)

3. Koji je položaj (koordinate) uključene diode na ekranu Micro:bit-a, prikazane na slici?



- A. (3,2)
- B. (2,3)
- C. (2,4)
- D. (4,2)

4. Pažljivo prouči kôd:



Šta će biti prikazano na ekranu Micro:bit-a?

- A. Uključene LED diode na (3,2) i (1,4).
- B. Isključene LED diode na (3,2) i (1,4).
- C. Uključene LED diode na (3,2) i (4,1).
- D. Neće biti ništa prikazano.

TAČNI ODGOVORI			
1 – C	2 – B	3 – A	4 – A

PRAĆENJE I PROCJENJIVANJE OBRAZOVNIH POSTIGNUĆA

Akcent obrazovno-vaspitnih ishoda koji su obuhvaćeni ovim priručnikom je na praktičnom znanju iz oblasti programiranja i elektronike, pa je suština aktivnosti u fazi provjere i ocjenjivanja znanja na radu sa praktičnim vježbama, a ne samo na teorijskim znanjima iz oblasti ICT-a. Brojčanom ocjenom (od 1 do 5) treba vrednovati praktične aktivnosti, ali i pokazano znanje i razumijevanje. Takođe, treba vrednovati kreativnost, originalnost rješenja, radno-tehničku spretnost, uvažavajući interesovanja, zalaganja i individualne sposobnosti učenika.

Primjeri kriterijuma ocjenjivanja:

V RAZRED		
NIVO POSTIGNUĆA		
Osnovni nivo	Srednji nivo	Viši nivo
<ul style="list-style-type: none"> - prepoznaje (uz pomoć nastavnika) i opisuje Micro:bit uređaj i radno okruženje programa MakeCode; - pokreće programe u MakeCode okruženju; - djelimično slaže blokove za izvršenje određenog zadatka; - nije dovoljno samostalan u primjeni znanja; - uključuje se u diskusiju samo uz podsticaj nastavnika. 	<ul style="list-style-type: none"> - vješto koristi tehniku slaganja blokova u cilju kreiranja određenog programa; - stvara uz pomoć nastavnika program u kojem se koristi ulaznim vrijednostima; - koristi blokove za čitanje podataka sa senzora; - povremeno se uključuje u diskusiju na času, a zadatke realizuje uz određene napomene. 	<ul style="list-style-type: none"> - samostalno stvara program koji se sastoji od slijeda koraka, ponavljanja i ulaznih vrijednosti; - analizira odnos ulaznih i izlaznih vrijednosti; - vrednuje uspješnost rješenja; - često postavlja pitanja i učestvuje u diskusijama; - izrađuje samostalno program; - procjenjuje ulogu senzora i dioda; - samostalno i na vrijeme završava zadatke, digitalni radovi su uredni, precizni i razumljivi, pomaže drugima.



PROJEKтна NASTAVA

Primjeri aktivnosti dati kroz projekte namijenjeni su za rad sa darovitim učenicima, članovima Informatičke sekcije (dodatnu nastavu ili Micro:bit klub). Podrazumijevajući da su učenici, članovi sekcija, prevazišli sadržaje redovne nastave, nastavnik može na osnovu lične procjene primijeniti pojmove i sadržaje planirane za stariji uzrast.

Programerska znanja učenika već su na dovoljno visokom nivou da mogu da ih upotrijebe u novom istraživačkom projektu. Učenici će, uz pomoć nastavnika, drugova i drugarica iz odjeljenja i roditelja, programirati Micro:bit. Temu projekta može odrediti nastavnik, a mogu je izabrati i sami učenici. Nastavnik može predložiti neki od naših projekata, pronaći na internetu odgovarajući ili sa učenicima osmisliti novi odgovarajući projekat.

PRVI ČAS

Učenici znaju da projekat nije aktivnost koja se dešava slučajno, već da ima svoju specifičnu strukturu. Podsjećamo ih na najvažnije elemente strukture.

Izbor teme i saradnika

- Razmislite o ponuđenim temama i izaberite jednu kojom ćete se baviti naredna tri školska časa.
- Pronađite saradnike – drugove i drugarice koji se interesuju za istu temu. Formirajte tim.
- Definišite projektni zadatak.
- I sl.

DRUGI ČAS

Projektne aktivnosti

- a. U okviru tima, definišite projektne aktivnosti.
- b. Odredite realizatore za svaku od aktivnosti.
- c. Odredite vremenski okvir za realizaciju aktivnosti.
- d. Predstavite rezultate aktivnosti (digitalne slike, tekstove i zvučne zapise) svim članovima grupe.

Prijedlog projekata

Dajemo vam primjere tri projekta koje možete da uradite. Projekte nećemo uraditi detaljno, ali su objašnjeni u dovoljnom obimu da steknete uvid u njihovu izradu. Ako želite, možete da nastavite da dorađujete izabrani projekat i na taj način kreirate novi.

Prepustite se mašti!



RAD NA PROJEKTU – „MJERENJE TEMPERATURE NA DALJINU“

Pokušaj da izmjeriš spoljašnju temperaturu na daljinu. Korišćenjem ovog principa možeš da napraviš i meteorološku stanicu.

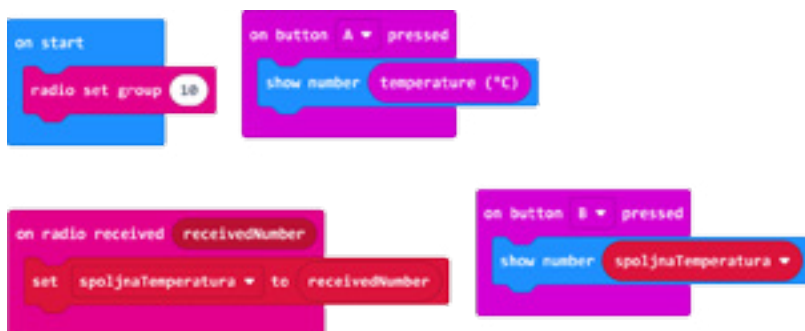
Šta je potrebno?

- dva BBC Micro:bit uređaja;
- pakovanje baterija (4 baterije);
- kesa ili plastična kutija (vodootporni materijal).

Kratak opis projekta

U ovom projektu koristimo dva različita programa, od kojih je jedan za Micro:bit na otvorenom, i prenosi vrijednost spoljašnje temperature korišćenjem radio-signala, u okviru jedne radio-grupe 10. Micro:bit koji se nalazi napolju koristi svoj temperaturni senzor za mjerenje temperature. Ovaj Micro:bit koristi radio-signal kako bi očitane temperaturu poslao Micro:bit-u koji se nalazi u unutrašnjosti. Kada Micro:bit koji je u sobi (ili zatvorenoj prostoriji) primi očitane vrijednost temperature spoljašnjeg Micro:bit-a, on treba da je prikaže na ekranu Micro:bit-a. Očitana temperatura se čuva u promjenljivoj koja je definisana pod imenom „spoljnaTemperatura“. Da bismo uporedili sobnu i spoljašnju temperaturu, postavljamo da, kada se pritisne taster A na Micro:bit-u koji je u sobi, prikazuje sobnu očitane temperaturu, dok kad je pritisnut taster B, na Micro:bit-u koji je u sobi prikazuje se temperatura koja je očitana sa spoljnog Micro:bit-a.

Kôd Micro:bit-a koji se nalazi u sobi: https://makecode.microbit.org/_gw1PsATeq7YV



Kôd Micro:bit-a koji se nalazi napolju: https://makecode.microbit.org/_3VYFL4EaD9Ud



RAD NA PROJEKTU – „POTAPANJE BRODOVA“

Napomena: Projekat „Potapanje brodova“ zahtijeva viši nivo znanja tako da se preporučuje za napredne učenike (npr. članove Informatičke sekcije ili članove Micro:bit kluba). Svrha realizacije ovog projekta je da kod učenika, kroz igru, probudimo interesovanje za sadržaje koje će usvajati u VI razredu.

Prati uputstva nastavnika i napiši kôd za igricu koja će da simulira poznatu igru „Potapanje brodova“. Da provjeriš tačnost kôda, odigraj sa drugarima igru, koristeći Micro:bit uređaje.

Šta je potrebno?

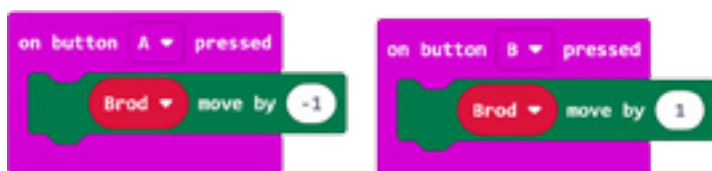
- jedan BBC Micro:bit uređaj;
- pakovanje baterija (2 baterije).

Kratak opis projekta

Učenici saznaju da je SPRAJT jedan piksel koji može da se kreće. Ima koordinate x i y, kojima se određuje smjer kretanja piksela po ekranu Micro:bit-a. Početna pozicija sprajta (broda).



Koristimo tastere A i B za kretanje piksela lijevo i desno.



Osim ovoga, koristićemo kombinaciju tastera A+B da upotrijebimo svoje oružje za pucanje. Prvo kreiramo promjenljivu „Oružje“ (kreiramo oružje koje nam služi za uništavanje neprijateljskih brodova).



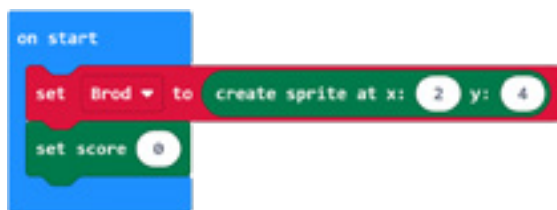
Neprijateljski brodovi dolaze sa gornje strane Micro:bit-a. Kreiramo promjenljivu „Neprijatelj“, čija je početna vrijednost definisana tako da x koordinata uzima nasumice vrijednosti od 0 do 4, a y koordinata ima vrijednost 0, jer neprijateljski brodovi napadaju sa vrha. Ubacujemo blok kojim provjeravamo da li je došlo do dodira sa neprijateljem.



Ujedno dodajemo i dio kôda kojim se provjerava da li je naš brod svojim oružjem uništio neprijatelja.



Da bismo omogućili da naša igra prikazuje skor, potrebno je dodati promjenljivu koja čuva rezultat.



Kompletan program možete vidjeti na linku: https://makecode.microbit.org/_JzTVWjUc6UYJ



RAD NA PROJEKTU – „PAZI, LOPOV!“

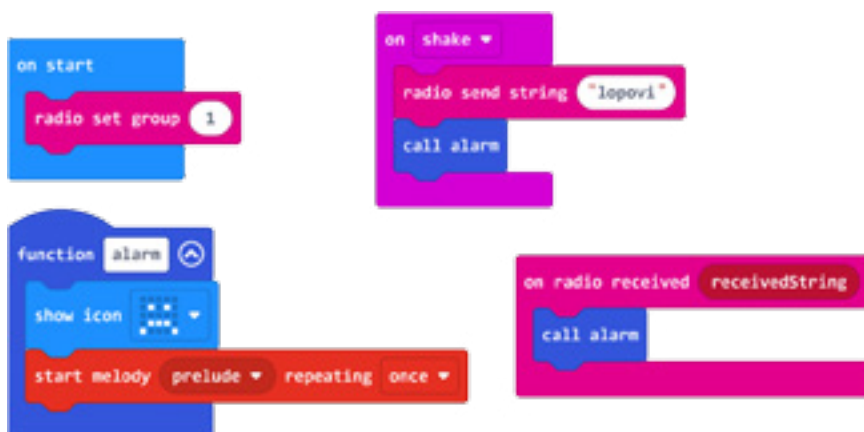
Napomena: Projekat „Pazi, lopov!“ zahtijeva veći nivo znanja. Potrebno je poznavati upotrebu funkcije, tako da se preporučuje za bolje učenike (npr. članove Informatičke sekcije). Pokušaj da napišeš program koji će ti pomoći da otkriješ moguću krađu tebi vrijedne stvari.

Šta je potrebno?

- dva BBC Micro:bit uređaja;
- pakovanje baterija (4 baterije);
- zvučnik.

Kratak opis projekta

Potrebno je da kreirate dva programa na dva Micro:bit-a. Jedan Micro:bit možete staviti u kutiju u kojoj se nalazi neka vama vrijedna stvar, a drugi, koji može biti povezan sa zvučnikom, postavite na neko vidno mjesto. Kada se kutija pomjeri, na Micro:bit-u unutar kutije akcelerometar detektuje pokret. Na ekranu Micro:bit-a se pokazuje ljuto lice. Takođe, šalje se poruka: „Lopov!“. Kada primi radio-poruku, drugi Micro:bit se oglašava melodijom, nalik sireni, i na taj način upozorava da je neko pomjerio predmet ili otvorio kutiju.



Program možete pogledati kada skenirate QR kôd ili na linku: https://makecode.microbit.org/_LYegHwEx340j



TREĆI ČAS

Predstavljanje rezultata projekta publici

- Napravite program u Bloksu poštujući sve četiri faze izrade programa.
 - Razmislite o problemu, osmislite rješenje.
 - U skladu sa idejom rješenja, odaberite i složite blokove.
 - Testirajte i analizirajte rad programa.
 - Ispravite greške (ako ih ima).
- Snimite video-zapis projekta.
- Osmislite scenario za usmenu prezentaciju.
- Podijelite uloge.

Lični osvrt na projektne aktivnosti

- Razmislite o problemima koji su se javljali i načinima na koje su prevaziđeni.
- Utvrdite koje odluke su bile dobre, a koje pogrešne.
- Da li je tim funkcionisao dobro ili je bilo određenih problema?

Procjena uspješnosti projekta

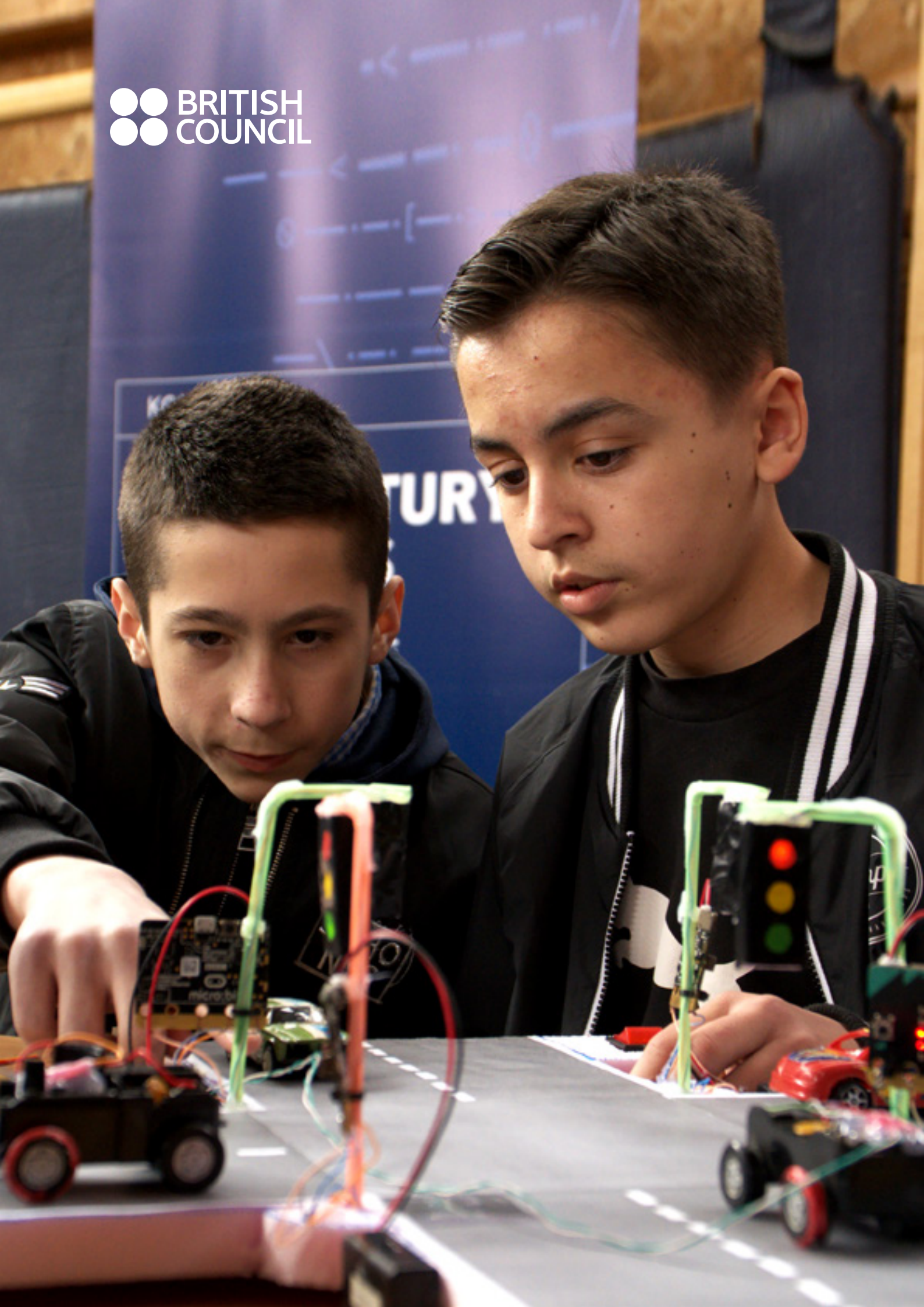
- Na osnovu povratnih informacija publike (drugova, drugarica i nastavnika) i ličnih zaključaka, u okviru tima procijenite uspješnost projekta.
- Formirajte listu zaključaka čijom primjenom bi projekat bio efikasnije realizovan.



OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA

Za realizaciju ishoda učenja iz programiranja u V razredu osnovne škole korišćeni su:

- projektor;
- računar (tablet ili pametni telefon);
- Micro:bit;
- mikro USB kabl;
- baterija sa 3 x AAA baterije;
- krokodilke štipaljke (Crocodile clip wires Micro:bit);
- zvučnik (Kitronik MI:Sound speaker board for BBC Micro:bit);
- materijal za kreativno izražavanje (ljepilo, ljepljive trake, folije, papir u boji, trake, žica...).





VI RAZRED

OBRAZOVNO-VASPITNI ISHOD 5

Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.

Ishodi učenja

Po završetku učenja učenik će moći da:

- razumije načine kreiranja i promjena vrijednosti varijabli;
- koristi aritmetičke operacije sa varijablama;
- razumije šta su uslovne naredbe i kada se koriste u programima;
- efektivno upotrebljava blokove *If...then* i *If...then...else* u programima;
- razumije šta je iteracija i kako se koristi u programima;
- efektivno upotrebljava petlje u programima.

Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda

a) Sadržaji/pojmovi:

Prilikom kreiranja promjenljive, zadavanja početne vrijednosti i promjene vrijednosti u programima, posebno naglasiti pravilno imenovanje varijabli; prilikom implementacije programa koji koriste cikluse *repeat*, *while* i *for*, koristiti primjere iz svakodnevnih aktivnosti učenika; za talentovane učenike izabrati kompleksnije zadatke, koji npr. koriste formule iz matematike. Pojmovi: promjenljiva, imenovanje varijabli, komentari, uslovne naredbe, *If...then* i *If...then...else*, iteracija, petlja (ciklus), ciklusi *repeat*, *while* i *for*, beskonačna petlja.

Napomena: Preporučujemo da se koristi okruženje za kodiranje koje kreira programe kombinovanjem blokova, bez pisanja samog kôda u tekstualnom obliku.

b) Aktivnosti učenja

Učenici:

- vježbaju pisanje programa u kojima kreiraju promjenljive, imenuju ih u skladu sa dobrom praksom, zadaju im početne vrijednosti i mijenjaju im vrijednosti;
- na papiru zapisuju aritmetičke operacije i prevode ih u odgovarajuće programe;
- uvježbavaju da pišu programe sa blokovima *If...then* i *If...then...else*;
- u grupama zapisuju primjere iteracije u svakodnevnom životu;
- vježbaju pisanje programa koji koriste cikluse *repeat*, *while* i *for*.

c) Broj časova realizacije (okvirno) – 7 časova.

*OBRAZOVNO-VASPITNI ISHOD 5 preuzet iz Plana i programa predmeta Informatika sa tehnikom.



DIDAKTIČKO-METODIČKA UPUTSTVA

Opšte didaktičke preporuke

Akcenat ovoga predmeta, sa posebnim osvrtom na ishode koji se odnose na programiranje, stavlja se na praktično znanje, pa je suština aktivnosti u učenju bazirana na praktičnim vježbama uz asistenciju nastavnika.

U skladu s dinamikom razvoja tehnoloških procesa i izvjesnih tehničkih inovacija, planirani nastavni sadržaji ovoga predmeta biće podložni izmjenama i dopunama. Zato ovaj priručnik, u njegovoj realizaciji, ostavlja slobodu svim učesnicima procesa.

Predviđeni sadržaji zahtijevaju raznovrstan metodički pristup, tj. obuhvataju korišćenje određenog programskog jezika, upotrebu raznih uređaja, upoznavanje i upravljanje raznim komponentama – ultrazvučni senzori, relej, servo motor, senzor pokreta, DC motora, foto-otpornika i sl.

Didaktičke preporuke za realizaciju OVI

Obrazovno-vaspitni ishod 5 realizovati kroz implementaciju što više programa. Insistirati na pravilnom imenovanju varijabli i jasnom razdvajanju blokova u kontrolnim blokovima; koristiti Micro:bit i druge raspoložive uređaje – komponente za demonstraciju programa.

**DIDAKTIČKO-METODIČKA UPUTSTVA preuzeta iz Plana i programa predmeta Informatika sa tehnikom.*



ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE

U ovom poglavlju dati su prijedlozi za realizaciju časova koji u potpunosti prate ishode definisane godišnjim planom i programom. U okviru svake oblasti skreće se pažnja na nove pojmove, čije se pojašnjenje nalazi i na kraju priručnika. Ova oblast je planirana kao pomoć nastavnicima u planiranju nastave, s tim što nastavnici časove mogu samostalno prilagođavati potrebama svog rada u svakom odjeljenju i u skladu sa svojim mogućnostima.

Takođe, planirani su i dodatni časovi u vidu projektne nastave, za čiju realizaciju je potrebno koristiti dodatnu opremu navedenu u pojedinačnim projektima.

PLAN ČASA BROJ 1

OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira i promijeni vrijednost promjenljive.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.

KRATAK OPIS ČASA

Kroz niz aktivnosti učenici kreiraju vrijednosti promjenljive, razumiju uvođenje promjenljive u programiranje – kao simboličkog imena za memorijsku lokaciju. Imenuju promjenljive, po sopstvenoj želji, uz preporuku da ih je najbolje imenovati tako da imena označavaju ono što u njih čuvamo. Na kraju časa se radi prezentacija svih radova, uz komentarisane i davanje sugestija.

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

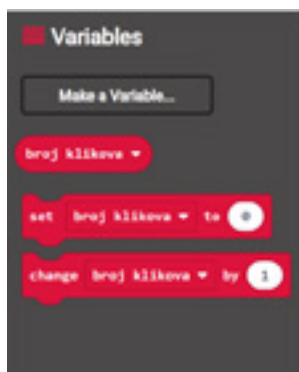
Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa pojmom promjenljive i njenom ulogom u programiranju.

Aktivnost 2

Zadatak – Nastavnik postavlja zadatak sa primjenom promjenljive u jednostavnom kôdu. Potrebno je da učenici kreiraju program koji klikom na taster A broji broj klikova i daje kao rezultat na ekranu Micro:bit-a broj klikova koji je korisnik napravio u realnom vremenu.

Aktivnost 3

Kreiranje promjenljive – Učenici, uz instrukcije nastavnika, otvaraju kategoriju *Variables*, zatim klikom na *Make a variable* kreiraju promjenljivu – dodjeljuju joj neko ime u zavisnosti od potrebe šta u njoj čuvaju. Kako je u ovom zadatku cilj da Micro:bit prikaže broj klikova na taster A, promjenljivu nazvati „broj klikova”.



Kreiranjem neke promjenljive dobijaju se tri nova bloka i to:

- definisanje imena promjenljive



- početna vrijednost promjenljive



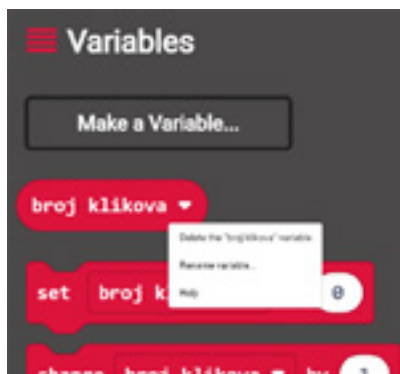
- promjena vrijednosti promjenljive



Aktivnost 4

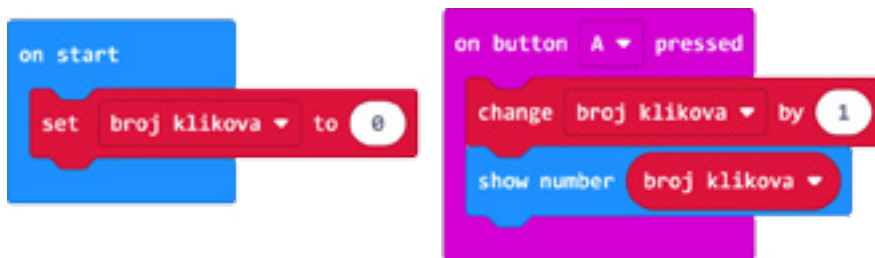
Promjena imena i brisanje promjenljive – Učenici se upoznaju sa načinom promjene imena i brisanja promjenljive, prateći instrukcije nastavnika. Dolaze pokazivačem miša do promjenljive koju su kreirali, zatim desnim klikom miša na promjenljivu dobijaju kontekstni

meni, koji nudi mogućnosti brisanja promjenljive, promjene imena ili pružanja pomoći. U zavisnosti od potrebe, koriste neku od opcija.



Aktivnost 5

Prezentacija radova – Učenici prezentuju kreirani kôd u kome su definisali promjenljivu „broj klikova”. Postavili su da početna vrijednost promjenljive „broj klikova” bude 0, a da se svakim klikom na dugme A vrijednost promjenljive „broj klikova” povećava za 1. Takođe, klikom na taster A na ekranu uređaja prikazuje se i trenutni broj klika koji je korisnik napravio. Učenici komentarišu dobijene rezultate i daju sugestije za eventualne korekcije u pisanju kôda.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_cKX4kab01daq



PLAN ČASA BROJ 2	
OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da koristi aritmetičke operacije sa promjenljivim.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.
KRATAK OPIS ČASA	
Kroz niz aktivnosti učenici kreiraju vrijednosti promjenljive i koriste matematičke operacije. Na ovom času učenici će kreirati kôd koji će množiti dva broja i na taj način provjeriti i svoje znanje tablice množenja. Pored množenja, u zavisnosti od vremenskih okvira, učenici će raditi na primjeni i drugih aritmetičkih operacija. Na kraju časa se radi prezentacija svih radova, uz komentarisanje i davanje sugestija.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

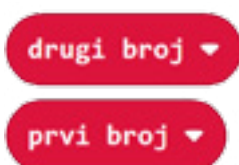
Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate uputstva nastavnika. Nastavnik upoznaje učenike sa zadatkom za čije rješenje je potrebno kreirati određeni kôd.

Aktivnost 2

Zadatak – Učenici kreiraju program koji klikom na taster A prikazuje kao rezultat na ekranu Micro:bit-a vrijednost prvog broja. Klikom na taster B prikazuje kao rezultat na ekranu Micro:bit-a vrijednost drugog broja. Dok se istovremenim klikom na taster A i B na ekranu Micro:bit-a prikazuje proizvod ta dva broja.

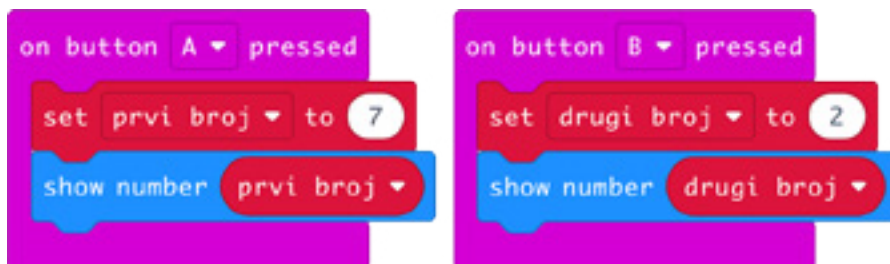
Aktivnost 3

Kreiranje promjenljive – Kako su se na prethodnom času učenici upoznali sa načinom kreiranja promjenljive, otvaraju kategoriju *Variables*, zatim klikom na *Make a variable* kreiraju dvije promjenljive koje mogu nazvati – „prvi broj” i „drugi broj”. Naravno, učenici mogu i sami definisati nazive promjenljive, po sopstvenom izboru.



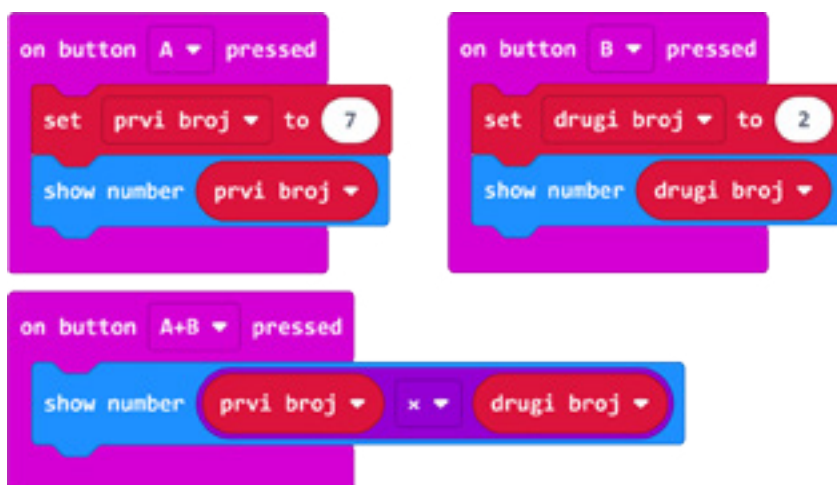
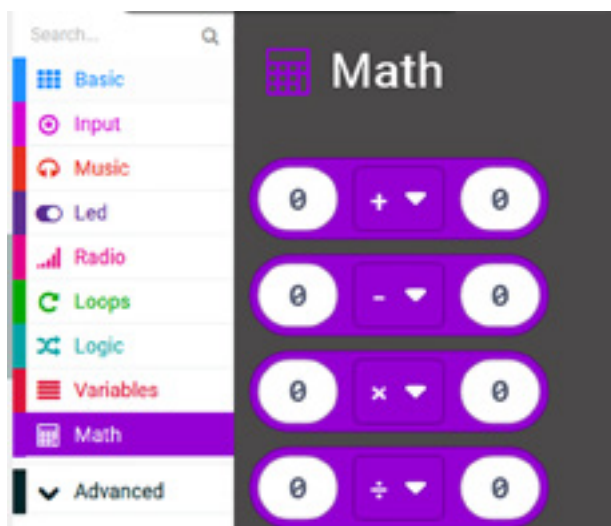
Aktivnost 4

Dodjeljivanje vrijednosti promjenljivama – Učenici iz kategorije *Input* prevlače na radnu površinu blok *on button A pressed* i u okviru njega iz kategorije *Variables* dodaju blok *set prvi broj to* i unose željeni broj u dijelu gdje se nalazi broj 0 (u ovom primjeru unose broj 7). Isti postupak se ponavlja za taster B i promjenljivu „drugi broj” kojoj dodjeljuju vrijednost 2.



Aktivnost 5

Matematičke operacije – Učenici iz kategorije *Input* prevlače na radnu površinu blok *on button A pressed* i iz padajućeg menija u zaglavlju bloka umjesto A biraju A+B. Ovaj blok predstavlja situaciju u kojoj su istovremeno pritisnuti tasteri A i B na Micro:bit-u. U okviru bloka *on button A+B pressed* dodaju iz kategorije *Basic* blok *show number* koji će omogućiti prikazivanje zbira dvije promjenljive. Iz kategorije *Math* uzimaju odgovarajući blok koji će im omogućiti sabiranje kreiranih promjenljivih. U ovom primjeru koriste blok za sabiranje, koji se nalazi u kategoriji *Math*.



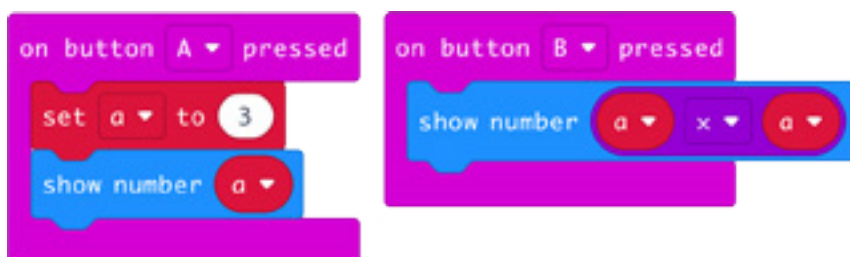
Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_A0KeCgc1HUja



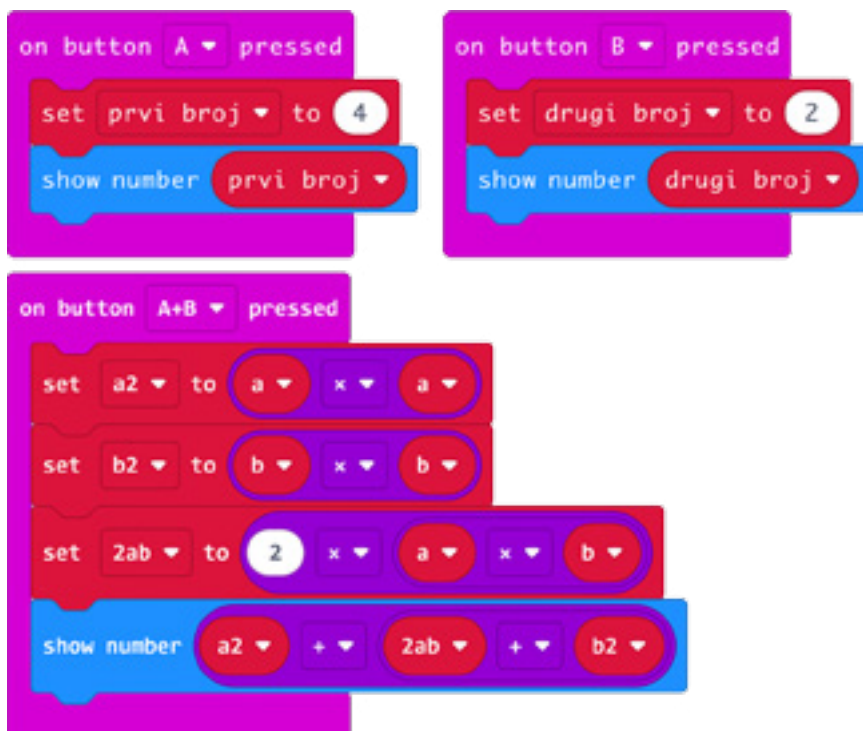
Aktivnost 6

Dodatni zadaci – Učenici kreiraju kôd za izračunavanje sljedećih izraza primjenom aritmetičkih operacija:

a) a^2



b) $a^2 + 2ab + b^2$



PLAN ČASA BROJ 3	
OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da objasni namjenu blokova <i>If...then</i> i <i>If...then...else</i> u programima.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.

KRATAK OPIS ČASA
Povezujući aktivnosti i obaveze iz svakodnevnog života, kao i donošenje odluka u vezi sa njima, učenici se upoznaju sa naredbom <i>If...then</i> i <i>If...then...else</i> . Za razliku od linijskih programa, gdje se svaki korak, odnosno svaka naredba, izvršava samo jedanput, u programima odlučivanja, tačnije razgranatim (uslovnim) programima, neke naredbe će se izvršiti, a neke ne, u zavisnosti od ispunjenosti uslova (odnosno donijete odluke).

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa namjenom blokova *If...then* i *If...then...else* u programima i objašnjava njihovo funkcionisanje.

Aktivnost 2

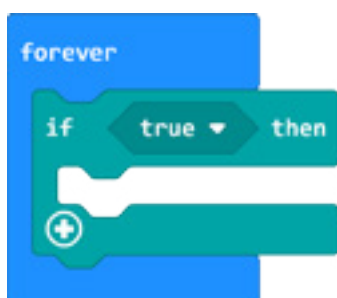
Zadatak – Učenici prate izlaganje nastavnika u postavci zadatka u kome je potrebno da naprave kôd koji će ako je temperatura u prostoriji $\geq 25^{\circ}$, na ekranu Micro:bit-a prikazati ↑ (strelicu gore), dok će u svim ostalim situacijama, tj. kada je temperatura $<25^{\circ}$, prikazivati ↓ (strelicu dolje).

Aktivnost 3

Senzor za mjerenje temperature – Micro:bit očitava stanje pomoću senzora temperature koji se nalazi u njegovom procesoru ili CPU-u. Temperatura procesora je približno jednaka temperaturi okoline u $^{\circ}\text{C}$ (celzijusu).

Iz kategorije *Basic* prevlače na radnu površinu blok *Forever* koji će im omogućiti da Micro:bit uvijek mjeri temperaturu i na taj način evidentira svaku moguću promjenu.

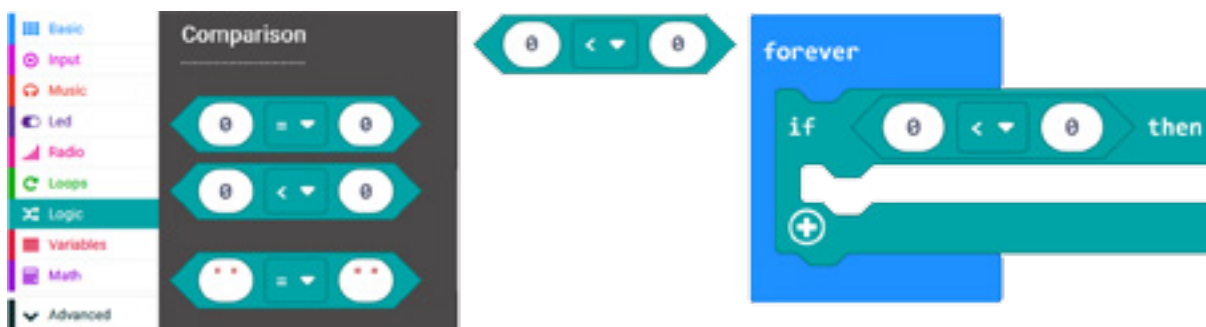
Iz kategorije *Logic* uzimaju blok *if... then* i dodaju ga u blok *Forever*. Blok *if... then* daje jednu od dvije vrijednosti, tj. tačno ili netačno. Kada je ulazna vrijednost tačna odnosno ispunjena (true), izvršiće se dio kôda koji slijedi poslije *then*. Kada je ulazna vrijednost netačna odnosno nije ispunjena, blokovi unutar *if... then* bloka neće se izvršiti.



Aktivnost 4

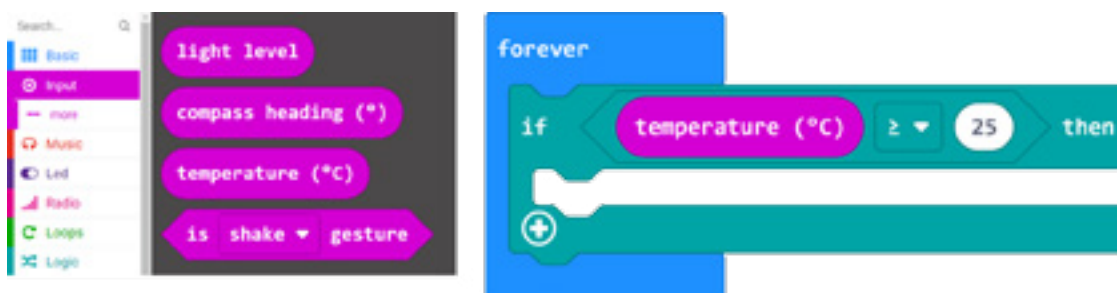
Definisanje uslova – Za potrebe definisanja uslova koji je dat u zadatku, tj. ako je temperatura u prostoriji $\geq 25^\circ$, na ekranu Micro:bit-a će se prikazati \uparrow (strelica gore), dok će u svim ostalim situacijama, tj. kada je $< 25^\circ$, prikazivati \downarrow (strelica dolje). Učenici koriste operatore poređenja koji se nalaze u kategoriji *Logic*.

Učenici iz palete ponuđenih blokova u okviru kategorije *Logic* uzimaju operator „ $0 < 0$ ” i iz padajućeg menija biraju \geq (veće ili jednako) koji služi za provjeru da li je vrijednost na lijevoj strani veća od vrijednosti na desnoj strani ili jednaka sa njom.



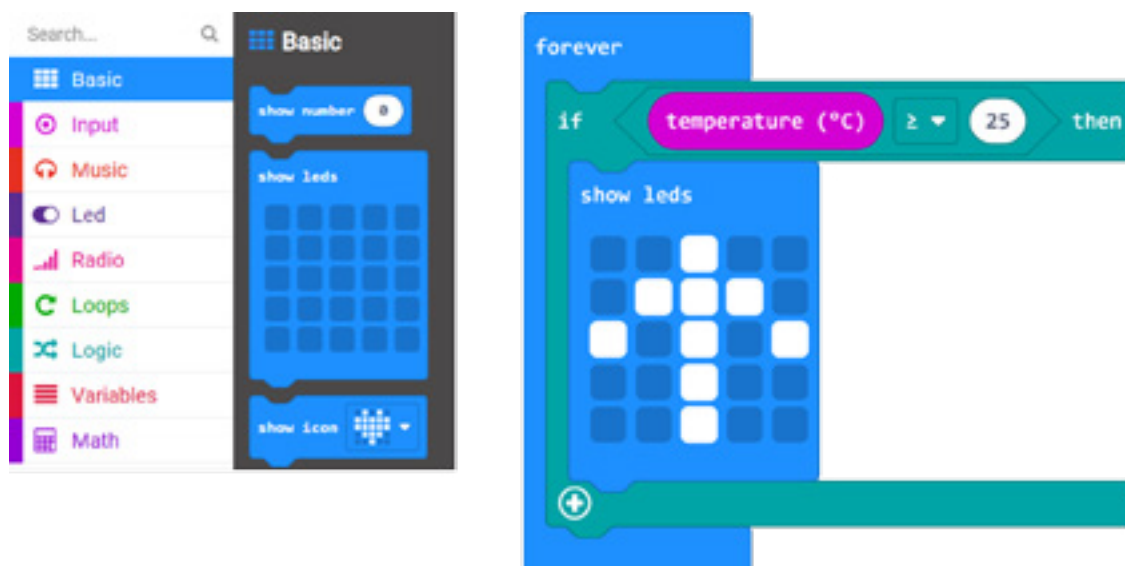
Očitavanje temperature – Blok koji očitava temperaturu naziva se *temperature (°)* i nalazi se u kategoriji *Input*.

Učenici prevlače mišem blok *temperature (°)* i postavljaju ga u prvo prazno polje operatora za ispitivanje uslova \geq , a u drugo polje unose vrijednost temperature definisanu zadatkom, tj. broj 25. Na ovaj način postavljen je uslov „ako je temperatura veća ili jednaka sa 25, onda” i nadalje treba definisati šta se dešava ako je uslov ispunjen, tj. ukoliko je True.



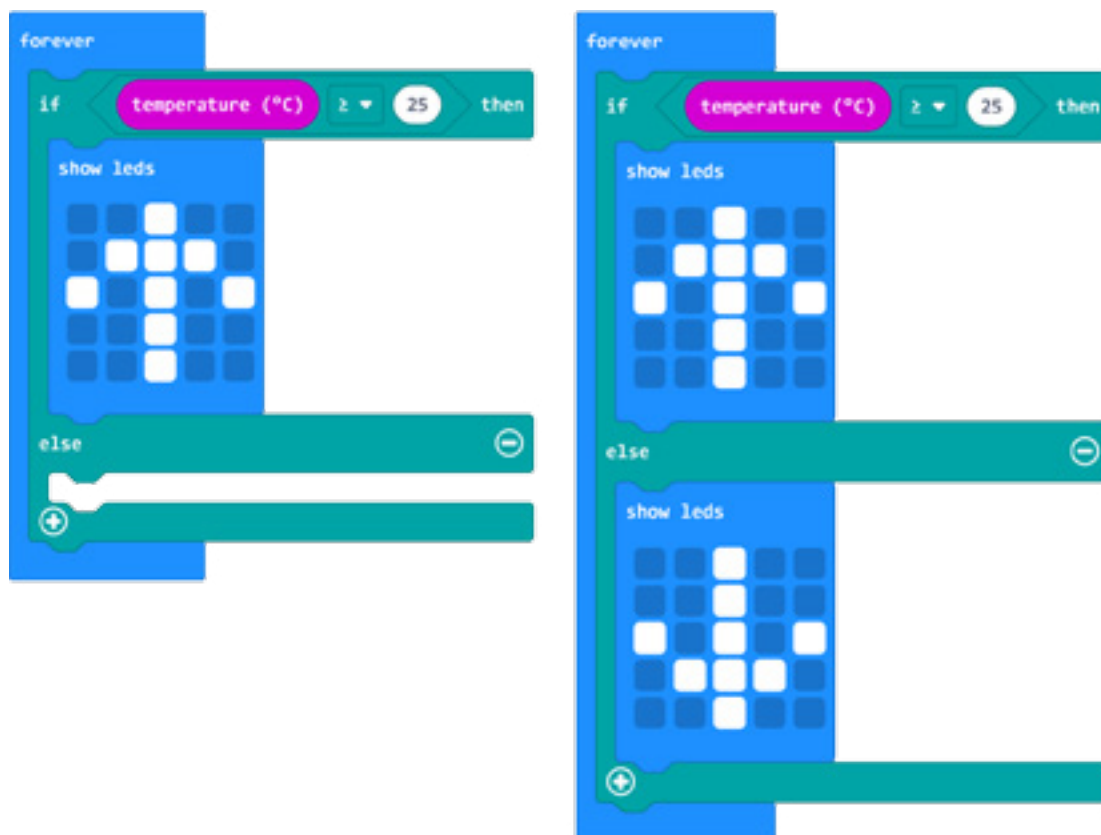
Aktivnost 5

Prikazivanje strelica – Na osnovu postavke zadatka, tj. ako je temperatura u prostoriji $\geq 25^\circ$, na ekranu Micro:bit-u treba prikazati \uparrow (strelicu gore), dok u svim ostalim situacijama, tj. kada je $< 25^\circ$, prikazivati \downarrow (strelicu dolje). Učenici u kategoriji *Basic* pronalaze blok *show leds*, prevlače ga na radnoj površini unutar bloka *if...then* i na njemu iscrtavaju \uparrow . Pokretanjem programa ispitaće se uslov i ukoliko je isti tačan, prikazaće se \uparrow .



Aktivnost 6

if...then...else – Kako je zadatkom definisano da u situaciji kada je temperatura $< 25^{\circ}$ treba prikazivati ↓, a u odnosu na postavljeni uslov ovo je suprotno, nije potrebno posebno definisati ovaj uslov već samo klikom na simbol plus (+) dodati sekciju *else* u trenutni *if...then* blok. Učenici unutar sekcije *else* postavljaju blok za prikazivanje ↓.



Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_fz42m3HV4hHd



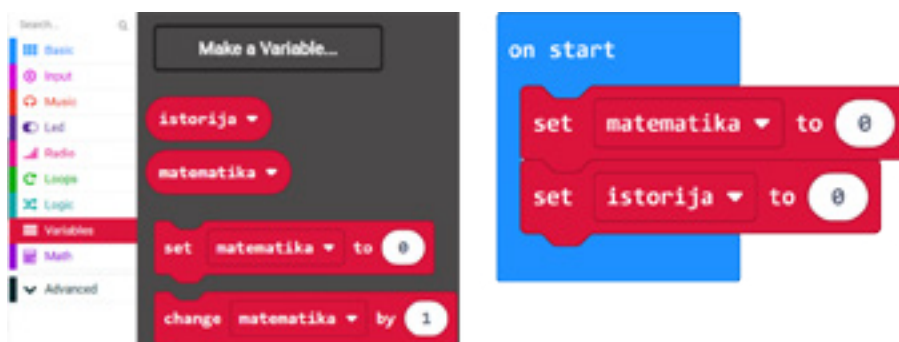
PLAN ČASA BROJ 4	
OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da koristi naredbe za grananje.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit.
KRATAK OPIS ČASA	
<p>Učenici izrađuju program koji će nakon „anketiranja” učenika dati odgovor na pitanje koji predmet učenici više vole. Predmeti koji će se naći u anketi su „Istorija i Matematika”.</p> <p>Anketiranje se vrši na način što učenik klikne na tastere A ili taster B, a sve u zavisnosti koji je njegov omiljeni predmet.</p> <p>Taster A će kliknuti svi učenici kojima je Matematika omiljeni predmet, dok će taster B kliknuti svi učenici kojima je Istorija omiljeni predmet. Po završetku izrade programa, svi učenici koji se žele anketirati treba da kliknu na jedan od tastera, a u skladu sa svojim afinitetima prema određenom predmetu.</p>	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

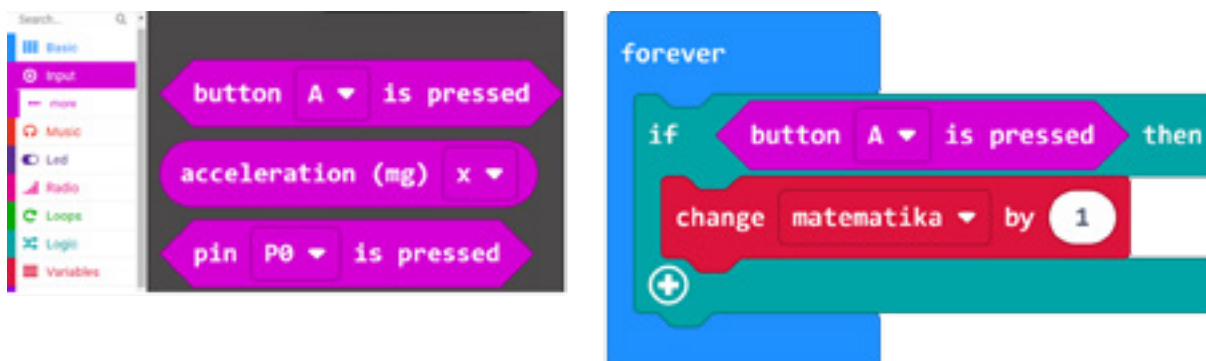
Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate instrukcije nastavnika. Nastavnik upoznaje učenike sa zadatkom i uvodi ih u rad. Učenici zaključuju da je za potrebe izrade zadatka potrebno kreirati dvije promjenljive koje će označavati predmete koji su ponuđeni. Na početku definišu dvije promjenljive „matematika” i „istorija”, postavljaju vrijednost promjenljivih na 0 u bloku *on start*.

Kako su učenici i ranije imali priliku da kreiraju promjenljive, pristupaju kreiranju istih.



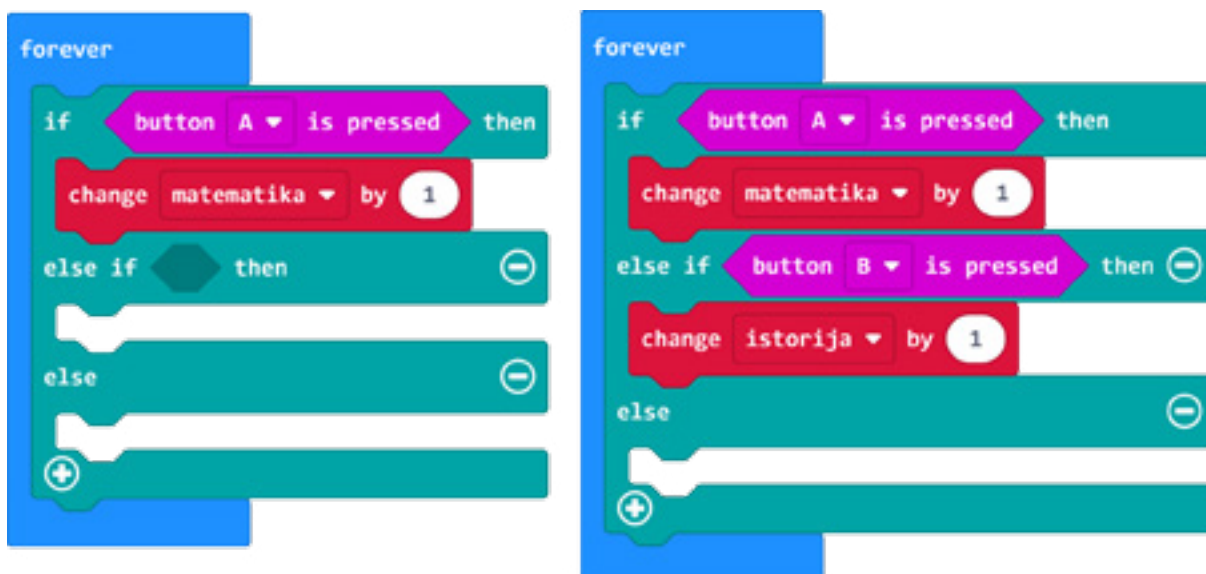
Aktivnost 2

Definisanje uslova – Učenici iz kategorije *Basic* uzimaju blok *forever* u koji dodaju blok *if...then* kako bi definisali uslove date zadatkom. Iz kategorije *Input* uzimaju blok *button A is pressed*, koji ima identičnu ulogu kao i ranije korišćeni blok *on button A pressed*, koji vraća True (tačno) ukoliko je kliknut taster A. Blok *button A is pressed* postavljaju u polje True. Ako je kliknut taster A, povećaj „promjenljive matematika” za 1.



Aktivnost 3

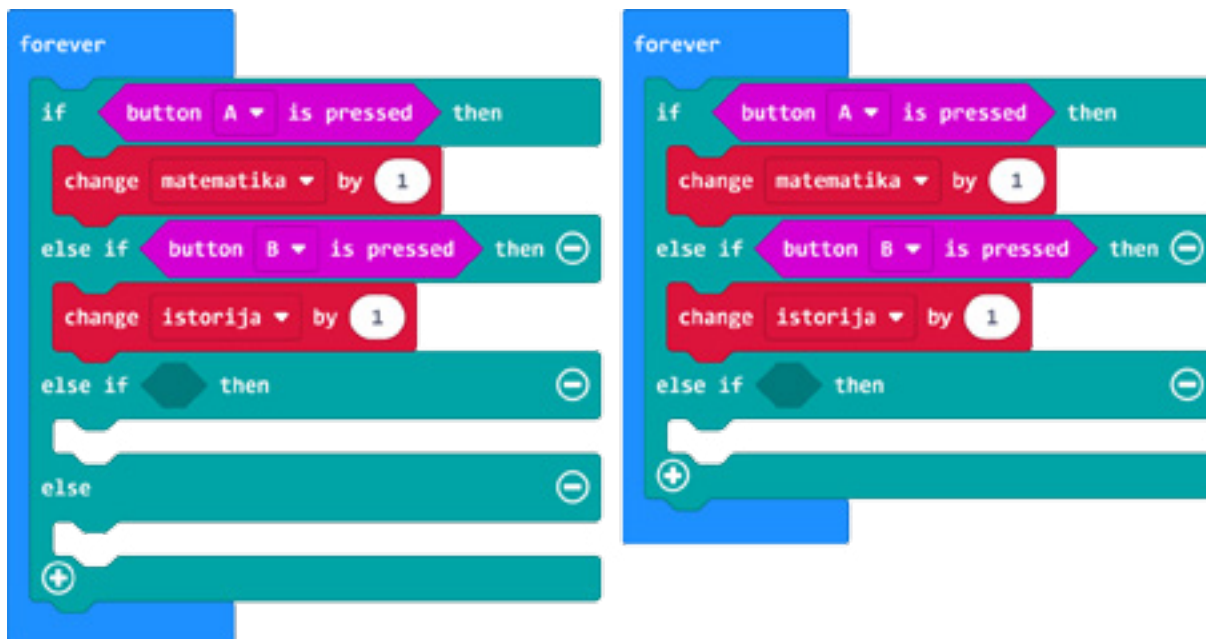
Definisanje uslova upotrebom *if...else if...else* – Kako su učenici i na prethodnom času imali priliku da se upoznaju sa dodavanjem sekcije *else*, sada to isto čine sa klikom na znak plus (+) i na taj način aktiviraju sekcije *else if* i *else*. Kako su zadatkom definisana tri stanja, tako i kroz blok *if...else* ispituju sva tri stanja. U okviru sekcije *else if...then* učenici definišu uslov „kliknuti na taster B, ako je istorija omiljeni predmet”.



Aktivnost 4

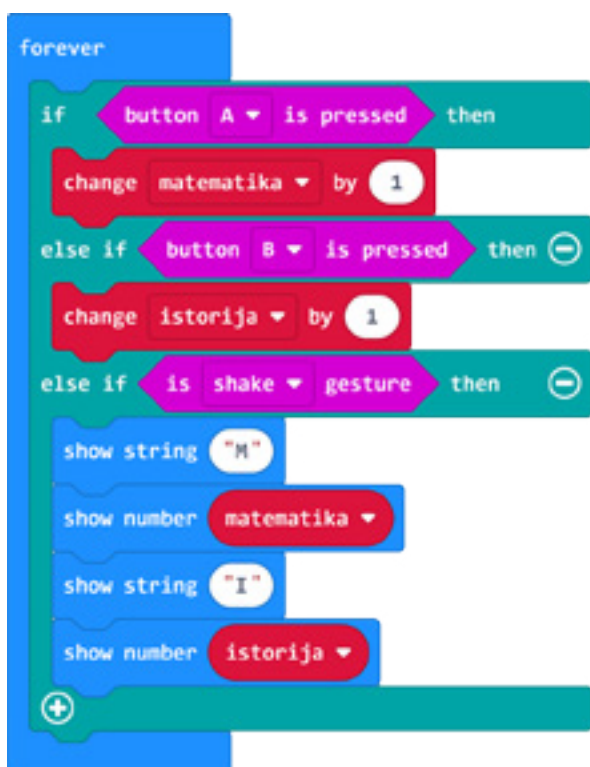
Definisanje uslova upotrebom *if...else if...else* – Ostalo je da ispituju još treće stanje, a to je „ako se Micro:bit protrese prikazati rezultate.” U ovoj situaciji ne možemo iskoristiti sekciju *else*, već moramo imati još jednu *else if...then*, koja će nam omogućiti precizno postavljanje uslova.

Da bi dobili još jednu sekciju *else if ... then*, učenici prvo kliknu na znak plus (+) na kraju bloka *if ... then*, nakon što se pojavi nova sekcija *else if...then* kliknu na znak minus (-) u produžetku sekcije *else*. Na ovaj način dobili su novu sekciju sa mogućnošću definisanja uslova, a uklonili sekciju *else* koja nije bila ni potrebna za ovaj kôd.



Aktivnost 5

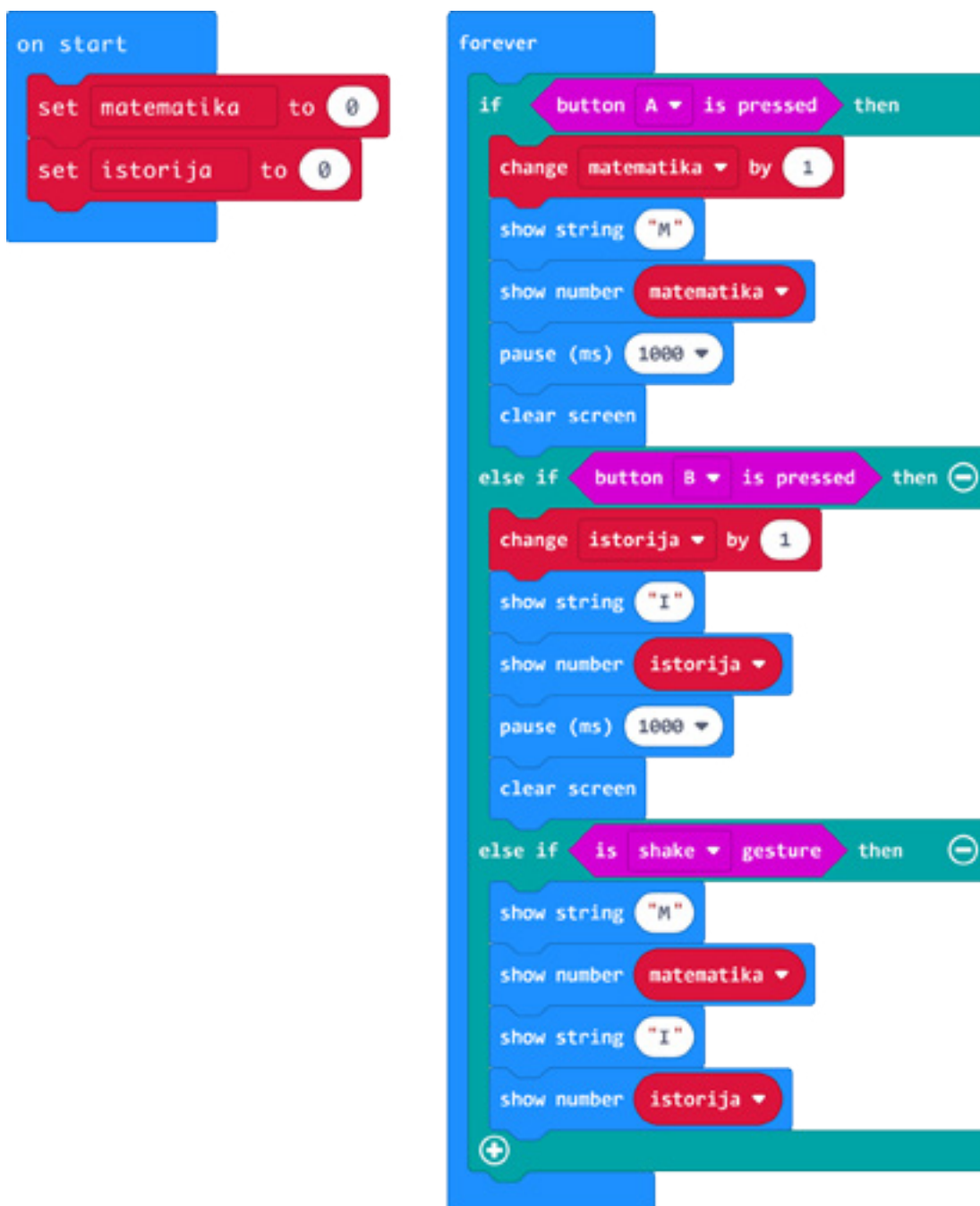
Prikazivanje rezultata – Učenici definišu treći uslov tako što dodaju blok *is shake gesture* iz kategorije *Input*. Ovaj blok ispituje da li se Micro:bit „protresao“. Ako jeste, to je stanje *True* i u tom slučaju izvršava se blok unutar ovog uslova, tj. prikazuju se rezultati.



Aktivnost 6

Prikazivanje rezultata – Učenici dodaju set blokova koji treba da prikažu rezultate ankete. Dodaju blokove *show string* i *show number*, koji će prikazati na ekranu Micro:bit-a sljedeće podatke: M (ukupni broj), I (ukupni broj).

Dakle, svaki put kada izaberu određeni predmet, tj. kliknu na A ili B, na ekranu Micro:bit-a će se prikazati počeno slovo predmeta i ukupan broj glasova u trajanju od 1 sekunde, zatim će se ekran obrisati.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_RgRL29VFtX8b



PLAN ČASA BROJ 5	
OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da primijeni blok <i>If...then...else</i> u programu za rad kompasa.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit.
KRATAK OPIS ČASA	
Učenici kreiraju program koji će u zavisnosti od smjera u kome je okrenut Micro:bit prikazivati na svom ekranu strane svijeta. Micro:bit u sebi sadrži magnetometar, dio koji mu omogućava da detektuje Zemljino magnetno polje, tj. čip koji omogućava da se odredi na koju stranu svijeta je okrenut Micro:bit. Kreiranjem programa i upotrebom blokova koji se odnose na detekciju Zemljinog magnetnog polja, Micro:bit možemo koristiti kao kompas.	

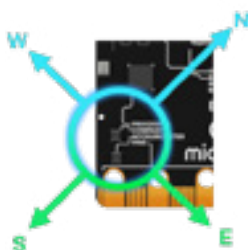
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa mogućnošću da Micro:bit odredi strane svijeta od 0 do 359 stepeni pomoću čipa magnetometra.

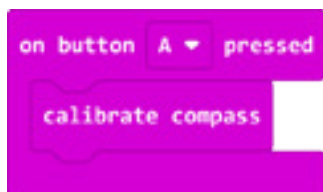
Aktivnost 2

Kompas – Učenici uočavaju da kompas detektuje Zemljino magnetno polje, što daje mogućnost određivanja smjera u kojem je okrenut Micro:bit. Čip magnetometra se nalazi na poleđini Micro:bit-a. Prilikom korišćenja Micro:bit-a kao kompasa, potrebno ga je usmjeriti prema sebi i gledati u njegov ekran (bilo da ga držite vertikalno ili horizontalno). Micro:bit će u odnosu na tu poziciju da odredi ugao i pokaže stranu svijeta.



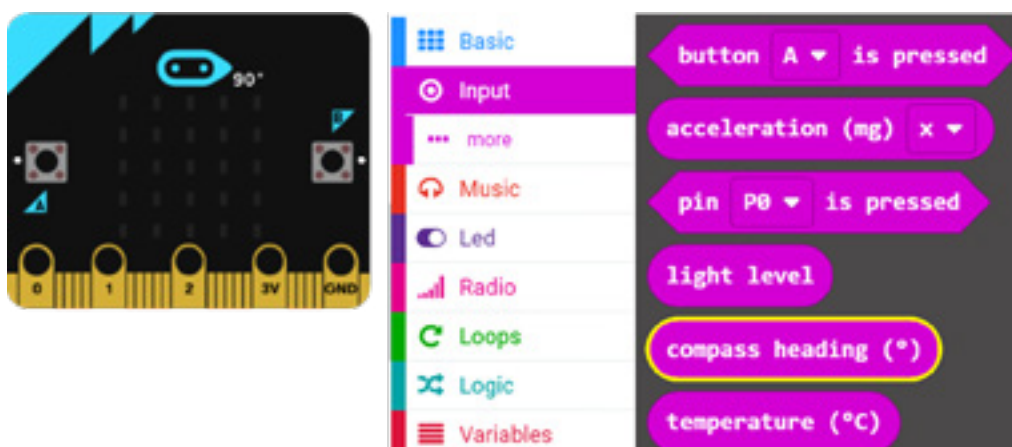
Aktivnost 3

Kalibracija – Učenici razumiju ulogu kalibracije i da se ona aktivira pokretanjem programa koji ima bilo koju od naredbi za aktivaciju kompasa. Na ekranu se pojavljuje poruka TILT TO FILL SCREEN. Zatim se Micro:bit mora rotirati u različitim smjerovima dok se sve LED diode na ekranu ne uključe i popune ga. Osmjeh će se tada prikazati kao znak da je kalibracija bila uspješna. Učenici kreiraju blok naredbi uz pomoć kojih klikom na taster A Micro:bit pokreće kalibraciju kompasa, tj. podešavanje. Uzimaju blok *calibrate compass*, koji se nalazi u kategoriji *Input* i smještaju ga u blok *on button A pressed*.



Aktivnost 4

Orijentacija kompasa – Prilikom korišćenja naredbe za smjer kompasa bloka *compass heading* (°), koja se nalazi u kategoriji *Input*, igla kompasa će se pojaviti na sredini vrha simulatora i ispisati stepene zavisno od toga kako je igla usmjerena. Učenici mogu rotirati iglu u oba smjera za 360° pomoću miša.

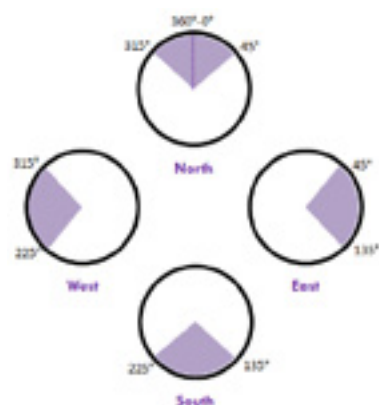


Aktivnost 5

Četiri strane svijeta² – učenici razumiju podjelu intervala na četiri strane svijeta i to na sljedeći način:

- sjever svi intervali $> 315^\circ$ i ≤ 360 ili $\geq 0^\circ$ i $\leq 45^\circ$;
- istok svi intervali $> 45^\circ$ i $\leq 135^\circ$;
- jug svi intervali $> 135^\circ$ i $\leq 225^\circ$;
- zapad svi intervali $> 225^\circ$ i $\leq 315^\circ$.

Na osnovu podjele pišu program za kompas.



² <https://www.101computing.net/bbc-microbit-digital-compass/cardinal-points/>

Aktivnost 6

Program – Učenici ispisuju program za kompas, definišu uslove i na osnovu istih ispisuju slovo jedne od četiri glavne strane svijeta na ekranu. Pretvaraju gore definisane uslove u programske blokove.

U *forever* petlju, nakon komande koja promjenljivoj „kompas” dodjeljuje vrijednost u obliku stepeni, provjeravaju u kom opsegu se nalaze detektovani stepeni kako bi se povezali sa ispravnom stranom svijeta.

Dodaju blok *if then* i u njemu naredbu *or* (ili) iz kategorije *Logic*. Ova komanda omogućava da testiraju dva uslova istovremeno, ali samo jedan od njih mora biti zadovoljen da bi se izvršile komande u ovom dijelu. Jedan uslov treba da provjeri da li je vrijednost promjenljive „kompas” veća od 315 i manja i jednaka sa 360, a drugi da li je veća ili jednaka sa 0 i manja i jednaka sa 45. Ako je jedan od ovih uslova tačan, neka se prikaže slovo „S” kao smjer sjevera.



Aktivnost 7

Program – Učenici nadalje ispisuju program koji vrši provjeru i za druge strane svijeta. Da bi dobili još jednu sekciju *else if ... then*, učenici kliknu na znak plus (+) na kraju bloka *if ... then*. Ako je izmjerena vrijednost veća od 45, ali manja ili jednaka sa 135, Micro:bit mora prikazati slovo „I” (oznaka za istok). Sada je potrebno koristiti naredbu *and* za provjeru oba uslova jer oba moraju biti ispunjena.

Isti postupak ponavljaju i za jug (J), samo mijenjaju vrijednosti stepeni i dodaju naredbu za ispis slova „J” (oznaka za jug).

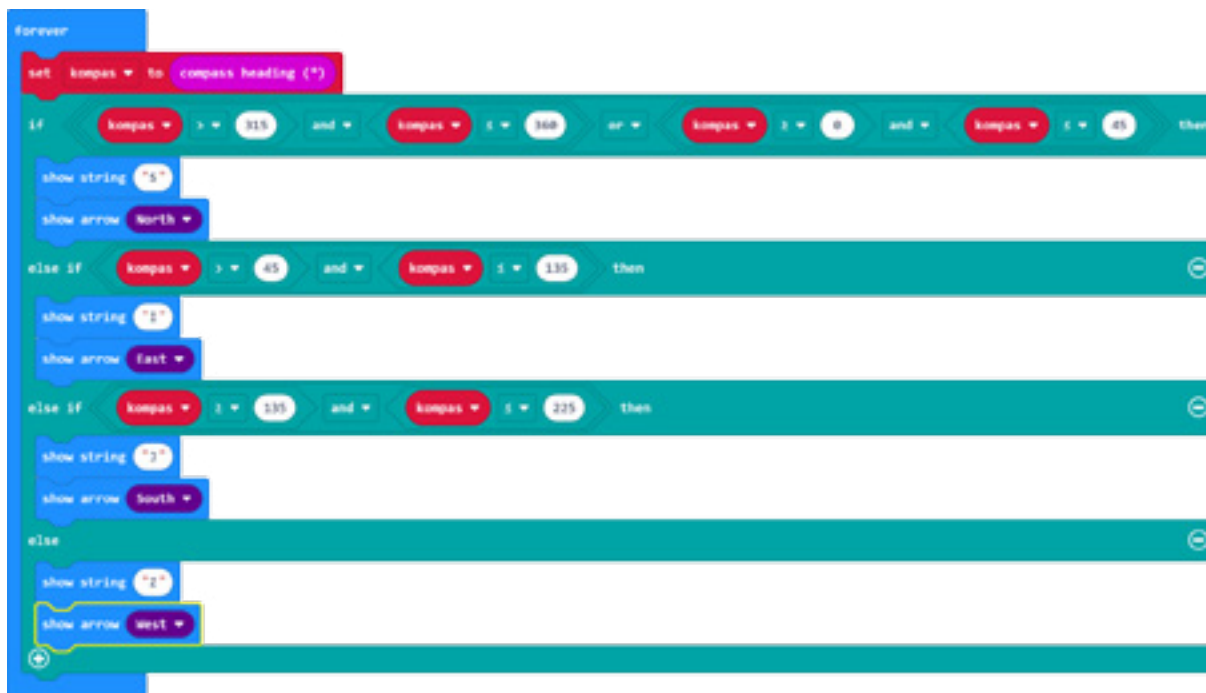
Kako je preostala još jedina opcija, opseg koji se odnosi na vrijednosti veće od 225 i istovremeno manje od i jednake sa 315, nije potrebno definisati ovaj uslov kao u prethodnim slučajevima. Stoga dodaju naredbu za ispis slova „Z” u drugom dijelu, jer je to oznaka za zapad.

Program dopunjavaju strelicama koje ukazuju na pravac određene strane svijeta.

Time je program završen.

Aktivnost 8

Testiranje – U simulatoru učenici pomjeraju iglu i prate prikaz slova na ekranu. Prebacuju program na Micro:bit i testiraju svoj kompas.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_UEXTd5RPrdYP



PLAN ČASA BROJ 6

OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da efektivno upotrebljava petlje u programima.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.

KRATAK OPIS ČASA

Kroz niz aktivnosti učenici rješavaju zadatak pronalaženja najvećeg zajedničkog djelioca dva broja, npr. 35 i 75. Ovaj zadatak je u korelaciji sa matematikom, jer učenici upravo u VI razredu izučavaju najveći zajednički djelilac (NZD). Za rješenje ovog zadatka nastavnik upoznaje učenike sa principom rada Euklidovog algoritma i primjenom istog rješavaju dati zadatak. Na kraju časa se radi prezentacija svih radova, uz komentarisanje i davanje sugestija.

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

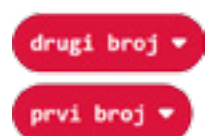
Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa principom rada Euklidovog algoritma.

Aktivnost 2

Zadatak – Učenici kreiraju program koji klikom na taster A prikazuje na ekranu Micro:bit-a neka dva broja, npr. 35 i 75. Klikom na taster B na ekranu će se prikazati najveći zajednički djelilac.

Aktivnost 3

Kreiranje promjenljive – Učenici otvaraju kategoriju *Variables*, zatim klikom na *Make a variable* kreiraju dvije promjenljive – „prvi broj” i „drugi broj”.



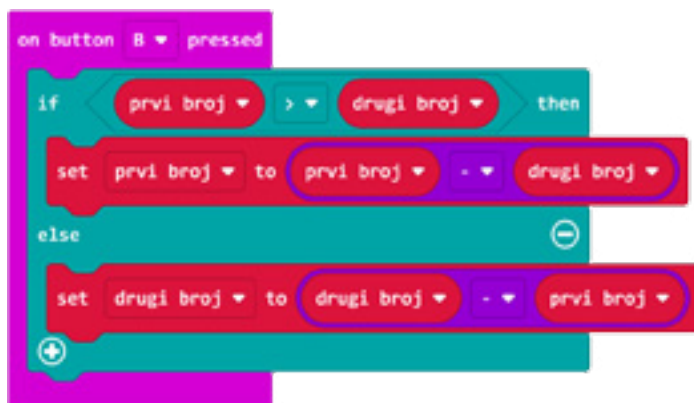
Aktivnost 4

Dodjeljivanje vrijednosti promjenljivama – Učenici dodjeljuju vrijednosti kreiranim promjenljivama. Za promjenljivu „prvi broj” upisuju vrijednost 35, a za „drugi broj” vrijednost 75. Prikazuju vrijednosti promjenljivih upotrebom *show number* bloka, zatim prave kratku pauzu od 1 sekunde između prikazivanja prvog i drugog broja.



Aktivnost 5

Euklidov algoritam – Učenici primjenjuju princip Euklidovog algoritma, koji se može opisati na sljedeći način: veći broj se umanjuje za vrijednost manjeg, novodobijene vrijednosti se porede i postupak se ponavlja sve dok se brojevi ne izjednače. Učenici definišu uslov „sve dok je prvi broj veći od drugog, od većeg broja oduzimaju manji”.



Aktivnost 6

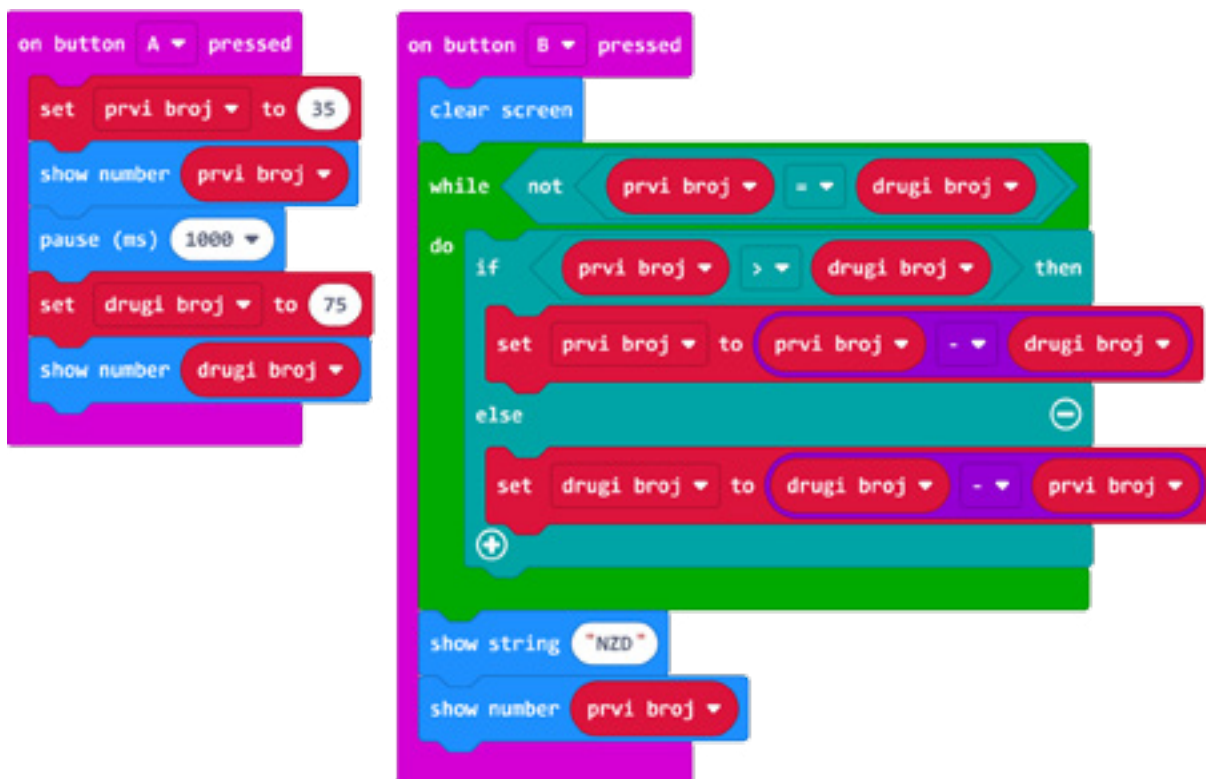
While petlja – Učenici pronalaze blok *while do* koji se nalazi u kategoriji *Loops*. Prevlače ga na radnu površinu u okviru bloka *on button B pressed*.

While petlja je dio kôda koji se izvršava dokle god je uslov te petlje zadovoljen. Dakle, kada dođemo do petlje u kôdu, računar provjerava da li je uslov ispunjen; ako jeste, on izvršava komande unutar petlje. Nakon što ih izvrši, vraća se ponovo do uslova; ako je on i dalje ispunjen, komande unutar petlje se ponovo izvršavaju. Sve se to ponavlja dok je uslov tačan, tj. petlja se prekida kada uslov postane netačan. Uslov koji definišu u okviru petlje *while...do* glasi „sve dok *prvi broj* nije jednak sa *drugi broj*”. To znači da će se unutrašnjost petlje izvršavati sve dok su brojevi različiti, a onog trena kada postanu isti, izlazi iz petlje i prikazuje taj broj. Broj koji se dobije predstavlja najveći zajednički djelilac dva zadata broja.



Aktivnost 7

Prezentacija radova – Učenici prezentuju svoj rad ostalim učenicima i zajedno komentarišu dobijene rezultate.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_VCbCgpYYi4ay



PLAN ČASA BROJ 7	
OVI 5	Na kraju učenja učenik će biti osposobljen da koristi promjenljive i kontrolne naredbe u programima.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da objasni namjenu bloka <i>for</i> u programu za ispisivanje parnih brojeva.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit.
KRATAK OPIS ČASA	
Učenici se kroz jednostavne programe upoznaju sa načinom korišćenja petlje <i>for</i> . Kreiraju program koji na ekranu Micro:bit-a prikazuje parne i neparne brojeve, brojeve djeljive određenim brojem i sl.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa petljom *for* u programima i objašnjava njeno funkcionisanje, tj. da se koristi u situacijama kada određenu naredbu ili niz naredbi treba izvršiti unaprijed zadani broj puta.

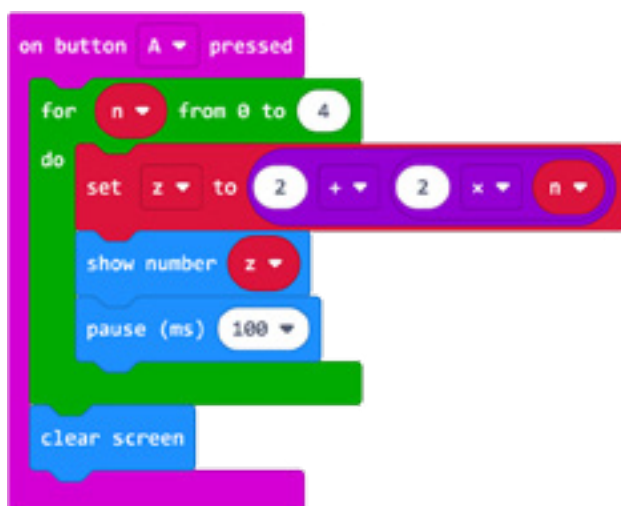
Aktivnost 2

Definisanje uslova – Učenici kreiraju program koji kao rezultat na ekranu Micro:bit-a prikazuje sljedeće:

- Pritiskom na taster A, na ekranu se ispisuju parni brojevi od 1 do 10.
- Pritiskom na taster B, na ekranu se ispisuju neparni brojevi od 1 do 10.
- Protresanjem Micro:bit-a, na ekranu se ispisuju svi brojevi od 1 do 20 koji su djeljivi sa brojem 3.
- Istovremenim pritiskom na tastere A i B, na ekranu se ispisuju svi brojevi od 1 do 20 koji su djeljivi sa brojem 5.
- Program je moguće pokretati neograničen broj ponavljanja.

Aktivnost 3

Parni brojevi – Učenici kreiraju program za ispisivanje parnih brojeva. Iz kategorije *Input* uzimaju naredbu *on button A pressed* iz kategorije *Loops* i unutar bloka *on button A pressed* dodaju blok *for*. Pritiskom na taster A, pokreću *for* petlju, koja će ispisati samo parne brojeve između 1 i 10. *For* petlja se sastoji od početne vrijednosti (0), promjenljive „n” koju koriste za čuvanje vrijednosti brojača, te krajnje vrijednosti (4). Prolaskom kroz petlju zadani broj ponavljanja (5), ispisuju se vrijednosti promjenljive „z” na Micro:bit-u (2, 4, 6, 8, 10). Blok *pause (ms)* definiše vrijeme (100) između prolaska kroz petlju u milisekundama (ms). Naredba *clear screen* briše sadržaj ekrana Micro:bit-a.

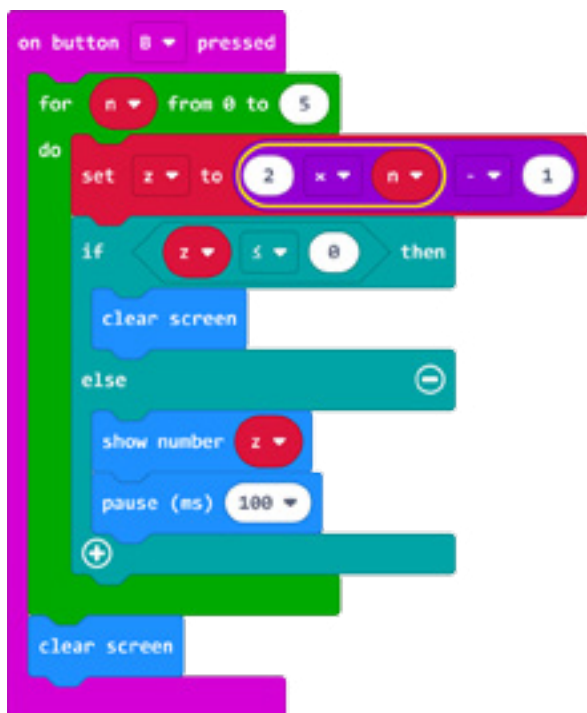


Aktivnost 4

Neparni brojevi – Učenici kreiraju program za ispisivanje neparnih brojeva. Pritiskom na tipku B, pokreće se *for* petlja, koja ispisuje samo neparne brojeve između 1 i 10. *For* petlja se sastoji od početne vrijednosti (0), promjenljive „n” koju učenici koriste za čuvanje vrijednosti brojača, te krajnje vrijednosti (5). Prolaskom kroz petlju zadani broj ponavljanja (6), ispisuje se

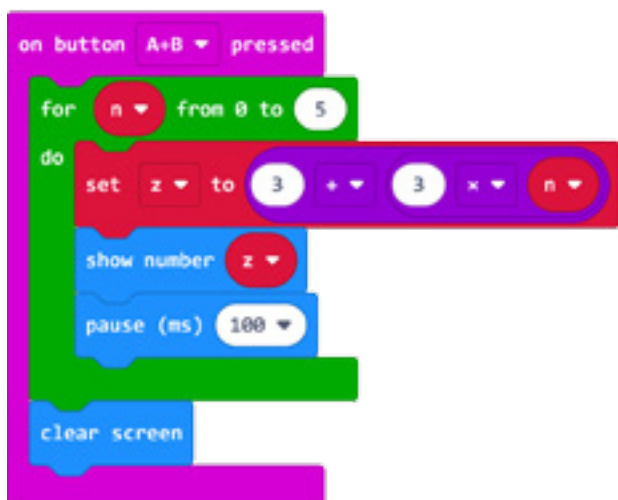
vrijednost promjenljive „z” na Micro:bit-u (1, 3, 5, 7, 9). Blok *pause (ms)* definiše vrijeme (100) između prolaska kroz petlju u milisekundama (ms).

Upotrebom bloka *if then...else*, učenici definišu uslov kojim se ne dozvoljava ispisivanje negativnih brojeva, tj. ako je vrijednost promjenljive „z” manja od 0, izbriši prikaz ekrana Micro:bit-a. Na kraju kôda, naredba *clear screen* briše sadržaj displeja Micro:bit-a.



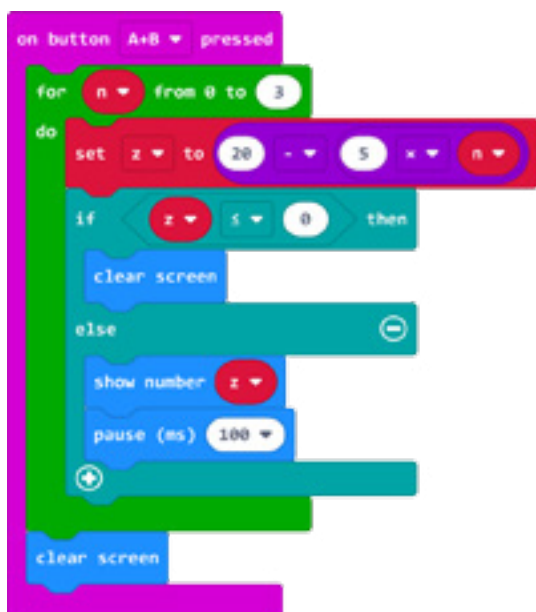
Aktivnost 5

Brojevi djeljivi sa 3 – Učenici kreiraju program za ispisivanje svih brojeva od 1 do 20 koji su djeljivi sa brojem 3. Iz kategorije *Input* biraju naredbu *on button pressed A*, klikom na strelicu pored A biraju A+B. Istovremenim pritiskom na dva tastera A i B pokrenuće se *for* petlja, koja će ispisati samo brojeve djeljive sa brojem 3. *For* petlja se sastoji od početne vrijednosti (0), promjenljive „n” koju učenici koriste za čuvanje vrijednosti brojača, te krajnje vrijednosti (5). Prolaskom kroz petlju zadani broj ponavljanja (6), na ekranu Micro:bit-a prikazaće se vrijednosti promjenljive „z” (3, 6, 9, 12, 15, 18). Blok *pause (ms)* definiše vrijeme (100) između prolaska kroz petlju u milisekundama (ms). Naredba *clear screen* briše sadržaj ekrana Micro:bit-a.



Aktivnost 6

Brojevi djeljivi sa 5 – Učenici kreiraju program za ispisivanje svih brojeva od 1 do 20 koji su djeljivi sa 5, i to od najvećeg prema najmanjem. Iz kategorije *Input* biraju naredbu *on shake*. Protresanjem Micro:bit-a, pokrenuće se *for* petlja, koja će ispisati samo brojeve između 1 i 20 djeljive sa 5, i to od najvećeg prema najmanjem. *For* petlja se sastoji od početne vrijednosti (0), promjenljive „n” koju učenici koriste za čuvanje vrijednosti brojača, te krajnje vrijednosti (3). Prolaskom kroz petlju zadani broj ponavljanja, na ekranu Micro:bit-a ispisivaće se vrijednost promjenljive „z” (20, 15, 10 i 5). Blok *pause (ms)* definiše vrijeme (100) između prolaska kroz petlju u milisekundama (ms). Kako unosimo samo pozitivne brojeve, naredbom *if then else* je zaustavljen *isps* kod broja 0. Naredba *clear screen* briše sadržaj ekrana Micro:bit-a.



Aktivnost 7

Testiranje – Nakon što su završili sa izradom programa, učenici na simulatoru provjeravaju izvršava li se program kao što su i predvidjeli. Kada su zadovoljni izrađenim programom, program prebacuju na svoj Micro:bit uređaj.



Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_cwRA98F7hi69



PRIMJERI PRAĆENJA PROCJENJIVANJA

Akcenat obrazovno-vaspitnih ishoda koji su obuhvaćeni ovim priručnikom je na praktičnom znanju iz oblasti programiranja i elektronike, pa je suština aktivnosti u fazi provjere i ocjenjivanja znanja na radu sa praktičnim vježbama, a ne samo na teorijskim znanjima iz oblasti ICT-a. Brojčanom ocjenom (od 1 do 5) treba vrednovati praktične aktivnosti, ali i pokazano znanje i razumijevanje. Takođe, treba vrednovati kreativnost, originalnost rješenja, radno-tehničku spretnost, uvažavajući interesovanja, zalaganja i individualne sposobnosti učenika.

Primjeri kriterijuma ocjenjivanja:

VI RAZRED		
NIVO POSTIGNUĆA		
Osnovni nivo	Srednji nivo	Viši nivo
<ul style="list-style-type: none"> - prepoznaje (uz pomoć nastavnika) i opisuje radno okruženje programa MakeCode; - pokreće programe MakeCode; - objašnjava pojam promjenljive; - djelimično slaže blokove za izvršenje određenog zadatka; - rijetko se uključuje u diskusiju i zadatke završava tek uz pomoć nastavnika. 	<ul style="list-style-type: none"> - samostalno ili uz pomoć nastavnika slaže niz blokova za rješenje problema, koristeći se naredbama ponavljanja i uslovima; - definiše pojam promjenljive i koristi je u izradi određenog zadatka; - koristi promjenljive i kontrolne naredbe u programima; - povremeno se uključuje u diskusiju na času, a zadatke realizuje uz određene napomene. 	<ul style="list-style-type: none"> - samostalno stvara program kao rješenje problema, koje uključuje niz naredbi primjenom algoritamske strukture; - samostalno razvija rješenje problema, koristeći se strukturom ponavljanja s određenim brojem ponavljanja; - koristi promjenljive i kontrolne naredbe u programima za rješavanje kompleksnijih izazova; - predviđa odgovarajuće ulazne (testne) primjere te kritički provjerava ispravnost rješenja i prema potrebi preuređuje svoje rješenje; - često postavlja pitanja i učestvuje u diskusijama; - samostalno i na vrijeme završava zadatke, digitalni radovi su uredni, precizni i razumljivi, pomaže drugima.



PROJEKTNNA NASTAVA

U Priručniku je dato mnoštvo primjera u kojima se pomoću MakeCode programa upravlja Micro:bit uređajem. Sve ideje su detaljno objašnjene, dok sam program i nije iz razloga što se očekuje da učenici imaju određeno predznanje u programiranju na programskom jeziku MakeCode. Primjeri dati u ovom dijelu namijenjeni su radu sa darovitim učenicima, kao i učenicima koji su članovi Kluba programiranja (sekcije).

Nakon jednostavnih početnih primjera, u kojima je objašnjena primjena osnovnih funkcija iz biblioteke za rad sa Micro:bit-om, u nastavku se nalaze i primjeri mini-projekata koje je moguće uraditi na za to predviđenim časovima.

Za rad sa Micro:bit-om i izradu projekata nastavnim programom predviđeno je sedam školskih časova u šestom razredu, pa je za šesti razred osnovna ideja da projekti koji se nalaze u Priručniku prvenstveno ukažu na raznovrsne mogućnosti Micro:bit-a i njegovu primjenljivost u rješavanju problema iz svakodnevnog života, a sve uz upotrebu dodatne opreme.

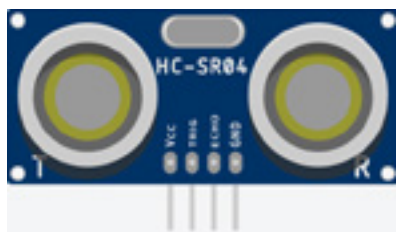
Uz projekte su dati, bez detaljnog uputstva, i prijedlozi nekih mogućih proširenja i nadogradnje, da bi se zainteresovanim učenicima pružila mogućnost da djelove projekta osmisle i ostvare samostalno. Ova proširenja su data i sa namjerom da inspirišu učenike i na originalne ideje i istraživački rad, kako bi sa svojim nastavnicima osmislili nova, kreativna projektna rješenja.

Za učenike posebno zainteresovane za programiranje, u dodatku su opisane neke od mogućnosti interaktivnog rada sa Micro:bit-om povezanim na personalni računar, a koje mogu biti vrlo korisne tokom izrade programerski zahtjevnijih projekata.



RAD NA PROJEKTU – „ULTRAZVUČNI SENZOR“

Ovaj projekat treba da omogući učenicima da razmotre sve situacije u svakodnevnom životu u kojima se susreću sa principom rada ultrazvučnog senzora. U ovom jednostavnom projektu ćemo prikazati osnovnu upotrebu senzora. Micro:bit će, pomoću ekrana, signalizirati da li se ispred senzora nalazi prepreka. Ukoliko detektuje prepreku na manje od 10 cm, na ekranu signalizira X. U suprotnom, prikazuje vrijednost distance.



Šta je potrebno?

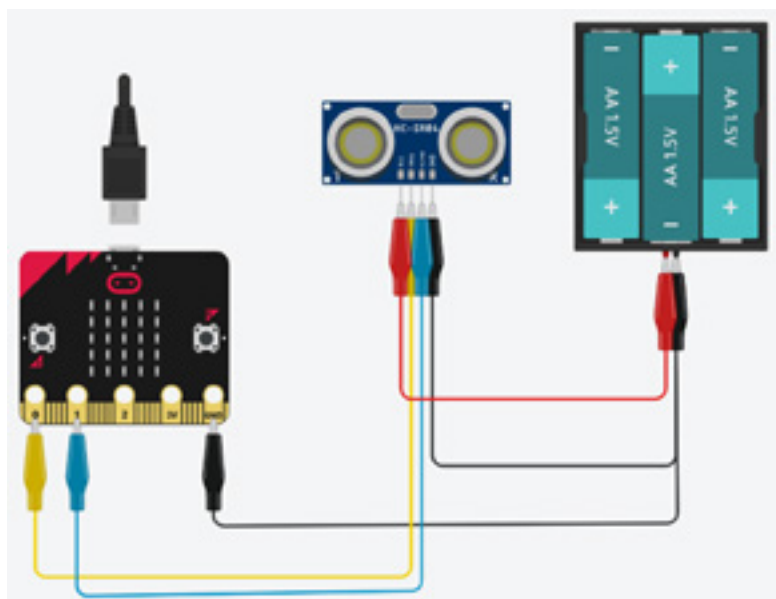
- Micro:bit;
- mikro USB kabl;
- ultrazvučni senzor za mjerenje udaljenosti (Ultrasonic Distance Sensor HC-SR04 5V);
- baterija sa 3 x AAA baterije;
- 5 x krokodil-kopče.

Razumijevanje ultrazvučnih senzora

Pojasnimo najprije ukratko šta je zvuk, a šta ultrazvuk. Talasi se kroz vazduh prostiru oscilovanjem čestica vazduha. Te oscilacije se prenose i na membranu u našem uhu. Kada je frekvencija ovog oscilovanja u određenom opsegu (do 20 KHz), onda to oscilovanje doživljavamo kao zvuk, tj. čujemo ga. Ultrazvukom nazivamo talase, odnosno oscilacije čija je frekvencija suviše visoka da bismo bili u stanju da ih čujemo. Dakle, zvučni i ultrazvučni talasi su po svojoj prirodi isti, a razlikuju se jedino po ljudskom doživljaju tih talasa (jedne čujemo, a druge ne).



Ultrazvučni senzor emituje talase visoke frekvencije (ultrazvučne talase). Princip rada ultrazvučnog senzora možemo objasniti kao slanje signala sa predajnika, odbijanje od eventualne prepreke i povratak tog istog signala ka prijemniku. Na osnovu vremena koje protekne između slanja i prijema odbijenog signala moguće je izračunati udaljenost između senzora i objekta.

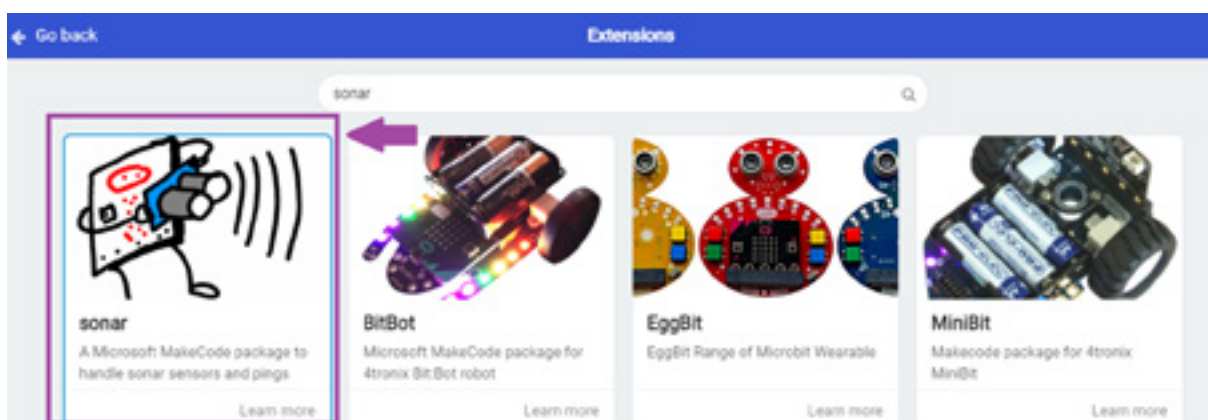
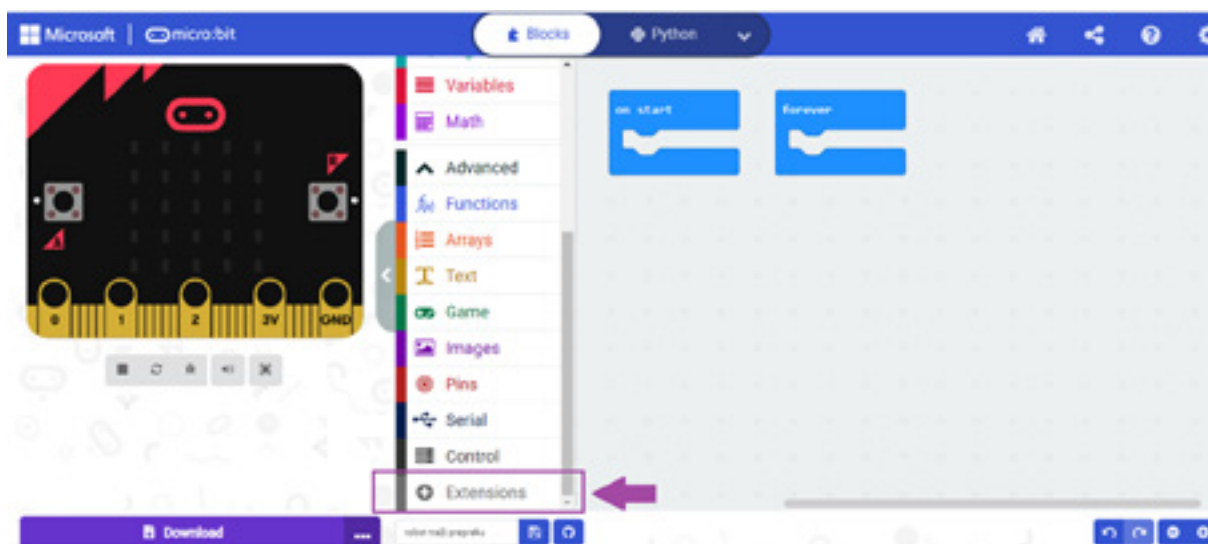
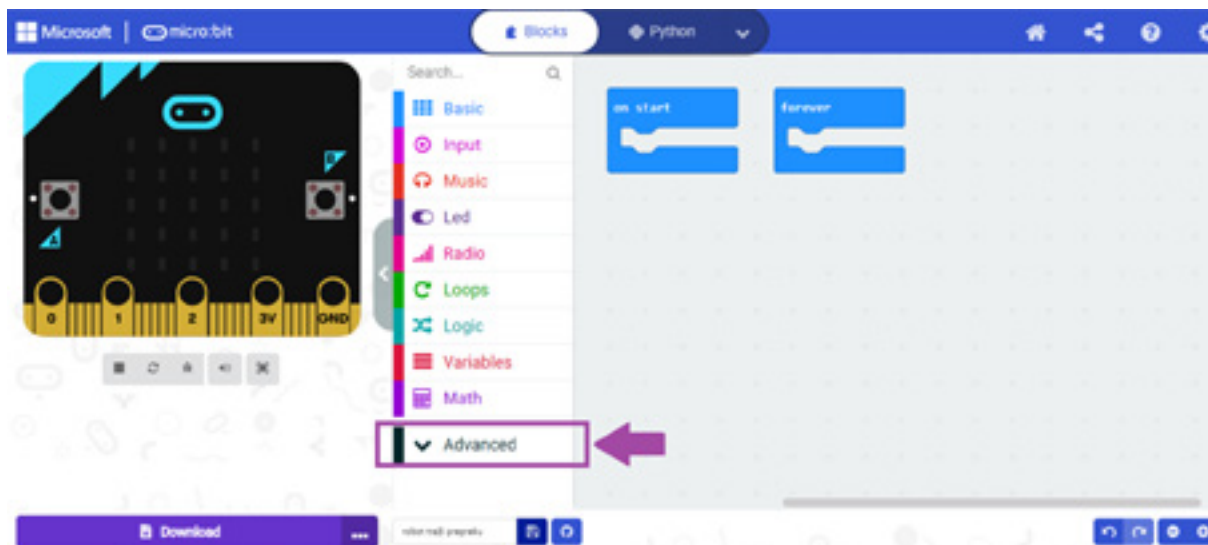


Ultrazvučni senzor radi na 5V i za njegov rad biće potrebno pakovanje od tri baterije od po 1.5V. Način povezivanja Micro:bit-a, ultrazvučnog senzora i baterija dat je na slici (iznad).

Na ultrazvučnom senzoru postoje četiri nožice: Vcc, Trig, Echo i GND. Pinovi Vcc i GND se povezuju tako da senzor dobije potreban napon (pri tome, GND pin treba povezati i sa GND kontaktom na Micro:bit-u, kao na slici). Trig i Echo pin možete povezati na pinove Micro:bit-a po želji (0, 1 ili 2), s tim da iste pinove upotrijebite i u programu.

Napisati program koji mjeri distancu i istu prikazuje na ekranu Micro:bit-a, a u slučaju da je distanca manja od 10, prikazuje X na ekranu Micro:bit-a.

Za potrebe izrade ovog zadatka, pored Micro:bit uređaja, biće vam potreban i ultrazvučni senzor, kao i dodatni blokovi koje moramo dodati klikom na *Advanced/Extensions/sonar*.



Prvo definišemo promjenljivu „distanca”, koja će čuvati sve vrijednosti. Postavljamo uslov – ukoliko je promjenljiva „distanca” manja od 10, program prikazuje X na Micro:bit-u. U suprotnom, prikazuje vrijednost distance.

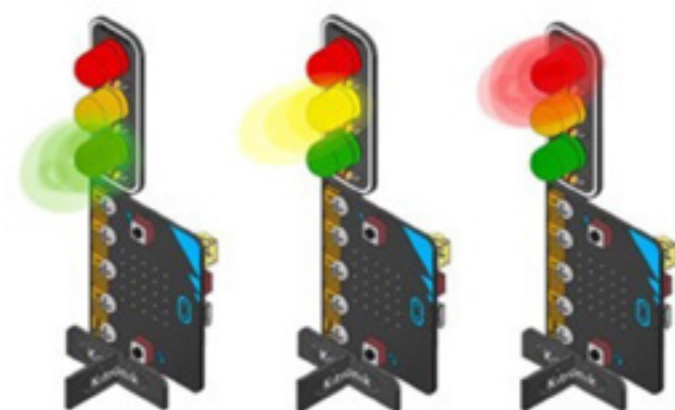


Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_M9oEogDHbaRb



RAD NA PROJEKTU – „SEMAFOR“

Ovaj projekat treba da ukaže na činjenicu da, iako već postoje standardizovana rješenja za određene probleme, tehnologija može postojeća rješenja učiniti još efikasnijim i praktičnijim. Napravićemo mali semafor i razmotriti koje sve mogućnosti savremeni semafor može da pruži u cilju boljeg obezbjeđenja učesnika u saobraćaju.

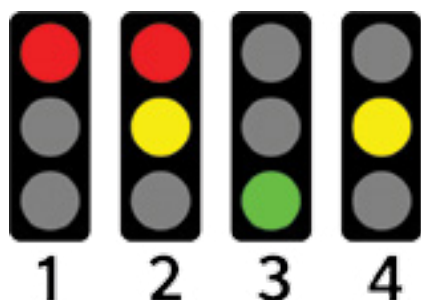


Šta je potrebno?

- Micro:bit;
- mikro USB kabl;
- semafor (Kitronik STOP:bit - Traffic Light for BBC Micro:bit);
- karton ili neki drugi materijal za izradu makete (po želji).

Napisati program koji će na semaforu emitovati LED lampice prema sljedećem rasporedu:

1. crveno svjetlo 10 sek.;
2. crveno+žuto svjetlo 3 sek.;
3. zeleno svjetlo 10 sek.;
4. žuto svjetlo 3 sek.



Povezivanjem Micro:bit-a sa komponentom semafor, koji je sastavni dio dodatne opreme i čije je povezivanje izuzetno jednostavno, tj. Micro:bit se naslanja na komponentu semafora tako da im se pinovi podudaraju, voditi računa da se GND poklopi sa GND. Upotrebom šarafa koji su takođe dio opreme vrši se povezivanje i direktna veza između Micro:bit-a i semafora.

Sada možemo uočiti da je P0 sa Micro:bit-a povezan na RED pin na semaforu, P1 na YEL, P3 na GRN. Blokovi koji nam omogućavaju upravljanje pinovima nalaze se u okviru *Advanced* u kategoriji *Pins*. Blok *digital write pin P0 to 0* mijenjamo kroz program u smislu što mijenjamo P0, P1 i P2 u zavisnosti od toga koju lampicu želimo aktivirati ili ugasiti.



Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_iYd2ku9J8476



OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA

- Micro:bit;
- mikro USB kabl;
- ultrazvučni senzor za mjerenje udaljenosti (Ultrasonic Distance Sensor HC-SR04 5V);
- baterija sa 3 x AAA baterije;
- krokodil-kopče;
- semafor (Kitronik STOP:bit - Traffic Light for BBC micro:bit).





VII RAZRED

OBRAZOVNO-VASPITNI ISHOD 5

Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.

Ishodi učenja

Po završetku učenja učenik će moći da:

- opiše opšti oblik funkcije, parametre funkcija i povratnu vrijednost;
- implementira funkcije sa i bez parametara;
- kreira programe koji koriste senzore;
- implementira programe koji koriste slučajne brojeve (*random numbers*);
- koristi grafički i tekstualni način zadavanja programa.

Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda

a) Sadržaji/pojmovi:

Prilikom pisanja programa koji koriste senzore, naglasak je na demonstraciji što većeg broja raspoloživih senzora (npr. kod Micro:bit uređaja akcelerometar, tilting, *measuring pitch, shaking, gestures*; senzori osvjetljenja...). Prilikom obrade funkcija, posebnu pažnju treba obratiti na funkcije sa i bez argumenata i na prelazak iz grafičkog predstavljanja blokova programa u tekstualni (pisanje kôda) i obratno. Pojmovi: senzori (npr. akcelerometar, tilting, *measuring pitch, shaking, gestures*), slučajni brojevi, funkcija, pozivanje funkcije, parametri (argumenti) funkcije, povratna vrijednost, grafički oblik programa, tekstualni oblik programa (kôd).

b) Aktivnosti učenja

Učenici:

- uvježbavaju pisanje programa u kojima očitavaju različite vrijednosti senzora;
- vježbaju implementaciju programa koji pozivaju generator slučajnih brojeva;
- zapisuju na papiru primjere kada je pogodnije koristiti funkcije;
- vježbaju pisanje funkcije bez argumenata i sa argumentima;
- vježbaju pisanje funkcije sa i bez povratne vrijednosti.

c) Broj časova realizacije (okvirno) – 7 časova.



DIDAKTIČKO-METODIČKA UPUTSTVA

Opšte didaktičke preporuke

Akcenat ovoga predmeta, sa posebnim osvrtom na ishode koji se odnose na programiranje, stavlja se na praktično znanje, pa je suština aktivnosti u učenju bazirana na praktičnim vježbama, uz asistenciju nastavnika.

U skladu s dinamikom razvoja tehnoloških procesa i izvjesnih tehničkih inovacija, planirani nastavni sadržaji ovoga predmeta biće podložni izmjenama i dopunama. Zato ovaj priručnik, u njegovoj realizaciji, ostavlja slobodu svim učesnicima procesa.

Predviđeni sadržaji zahtijevaju raznovrstan metodički pristup, tj. obuhvataju korišćenje određenog programskog jezika, upotrebu raznih uređaja, upoznavanje i upravljanje raznim komponentama – ultrazvučni senzori, relej, servo motor, senzor pokreta, DC motora, foto-otpornika i sl.

Didaktičke preporuke za realizaciju OVI

U obrazovno-vaspitnom ishodu 5 treba pokazati rad svih dostupnih senzora, objasniti razloge uvođenja funkcija i demonstrirati programe koji koriste funkcije sa i bez argumenata. U ovom ishodu učenici se prvi put susreću sa tekstom programa (kôdom) jer su u V i VI razredu upotrebljavali vizuelne alate za kreiranje programa. Treba koristiti uređaj Micro:bit i druge slične uređaje.

**DIDAKTIČKO-METODIČKA UPUTSTVA preuzeta iz Plana i programa predmeta Informatika sa tehnikom.*



ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE

Prilikom pripreme materijala za realizaciju nastave, nastavnik treba da izbjegava klasična predavanja, koja se sastoje od detaljnih, predugih i/ili dosadnih instrukcija. Poželjno je kreirati scenario časa tako da učenici imaju dosta mogućnosti da aktivno učestvuju u procesu učenja. Važno je da učenici mogu da primijene i vježbaju nove vještine, jer to dovodi do veće motivacije za učenje i veće podrške u primjeni novih znanja u njihovom radnom/životnom okruženju.

Zadatak nastavnika je da ohrabruje učenike, da im daje više vremena i prostora da u toku časa vježbaju novostečena znanja i vještine, uz primjenu prateće opreme koja je dostupna u školama. Od obrazovnog ishoda i ishoda učenja zavisi i to kako će nastavnik pripremiti scenario za čas ili časove.

Za izradu scenarija nastavnik treba da pođe od obrazovnih ishoda i ishoda učenja koji se odnose na programiranje, vodeći pritom računa o ciljevima predmeta, kako saznavnih tako i procesnih. Nakon postavljanja obrazovnih ishoda, pravi izbor odgovarajućih resursa za učenje, osmišljava adekvatne aktivnosti učenja i izbor načina ocjenjivanja. Zatim prelazi na konkretan ishod ili ishode učenja i strategije učenja, odnosno planira aktivnosti učenika koje vode ka ostvarivanju ishoda učenja. Na kraju planira mehanizme za provjeru postignuća.

Aktivnosti učenja su ključne za ostvarivanje obrazovnih ishoda programa i od toga kakve su te aktivnosti zavisi i kvalitet nastavnog procesa. Za planiranje kvalitetnog procesa učenja ključno je uspostavljanje logičkog odnosa između ishoda učenja, sa jedne, i aktivnosti učenja, sa druge strane.

PLAN ČASA BROJ 1	
OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik razumije mogućnosti funkcija Micro:bit-a (Micro:bit Functions), opšti oblik funkcije, kao i funkcije sa i bez argumenata (parametara).
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.

KRATAK OPIS ČASA
<p>Učenici će se upoznati s Micro:bit funkcionalnostima za kreiranje funkcija i njihovo pozivanje unutar nekog bloka. Za potrebe učenja, kreiraćemo jednostavnu termometar aplikaciju sa mogućnošću konvertovanja temperaturnih jedinica celzijus i farenhajt.</p> <p>Algoritam za kreiranje konvertera je sljedeći:</p> <ol style="list-style-type: none"> 1. Pomoću integrisanog termometra izmjerićemo trenutnu temperaturu i tu vrijednost čuvati u jednoj promjenljivoj. 2. Klikom na dugme A, program će pozvati funkciju koja nema parametre, već ispisuje trenutno izmjerenu temperaturu u predefinisanoj mjernoj jedinici – celzijusima. 3. Klikom na dugme B, program će kao parametar uzeti promjenljivu koja čuva vrijednost izmjerene temperature i pretvoriti je u farenhajte. Formula za konverziju je $t = 1,8 * t + 32$.

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike s konceptom funkcija i njihovom namjenom, te načinom na koji se koriste u MakeCode okruženju.

Aktivnost 2

Kreiranje promjenljivih – Učenici kreiraju promjenljivu u *forever* petlji kako bismo u svakom momentu imali tačnu temperaturu i dodijelili joj vrijednost koristeći integrisani termometar.

Aktivnost 3

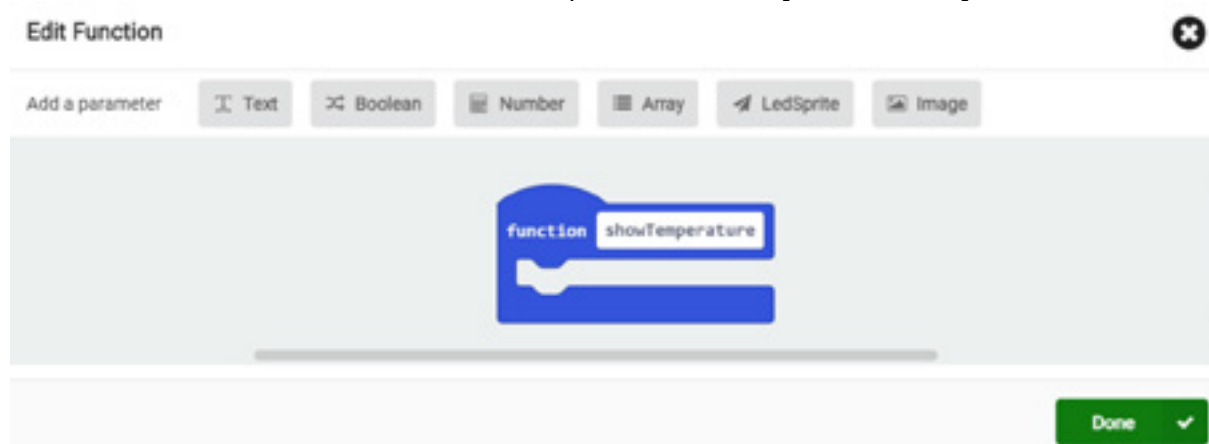
Kreiranje funkcija za A i B dugme – Učenici će potom kreirati dva bloka, odnosno dvije predefinisane mogućnosti korišćenja dugmeta A i dugmeta B na Micro:bit-u.



Aktivnost 4

Kreiranja *showTemperature* funkcije – Učenici su ranije upoznati s idejom da se na klik dugmeta A samo prikazuje temperatura u celzijusima pozivom funkcije koju smo nazvali „*showTemperature*”. Funkciju ćemo kreirati korišćenjem Functions funkcionalnosti:

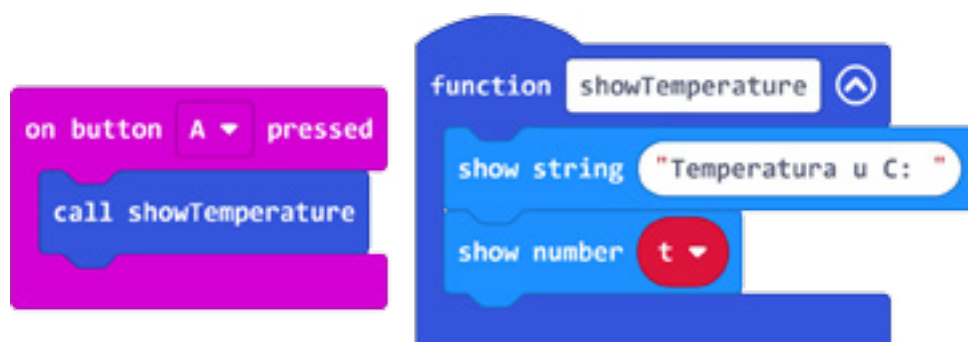
Odabirom *Make a function* otvoriće nam se prozor za kreiranje nove funkcije.



Aktivnost 5

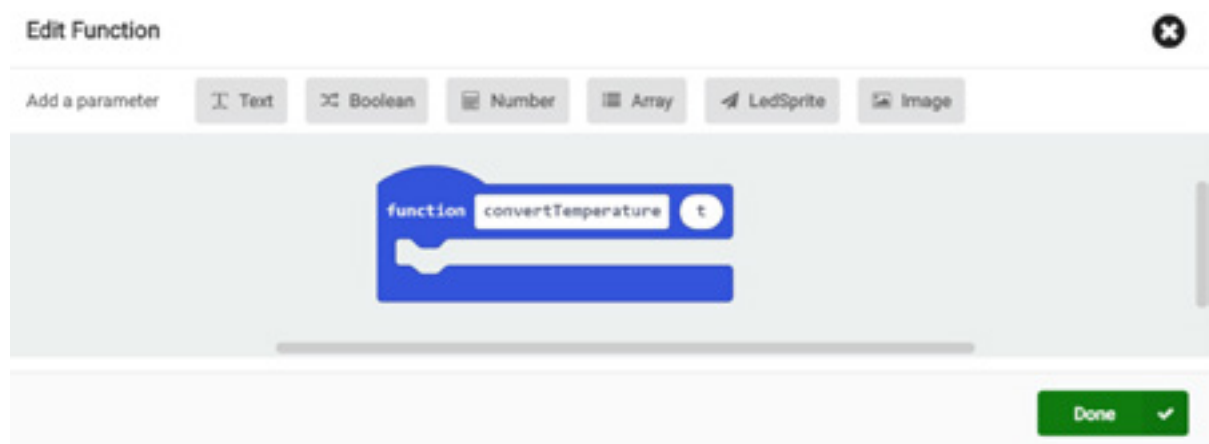
Poziv funkcije – Ponovnim pregledom *Functions* panela, učenici će primijetiti da se kreirala funkcija koja se može pozvati sa *call showTemperature*.

Učenici će zatim poziv ove funkcije dodati u blok za klik na dugme A (*on button A pressed*). Osim toga, unutar funkcije „showTemperature” unijećemo string „Temperatura u C”: dodati prikaz promjenljive „t”.



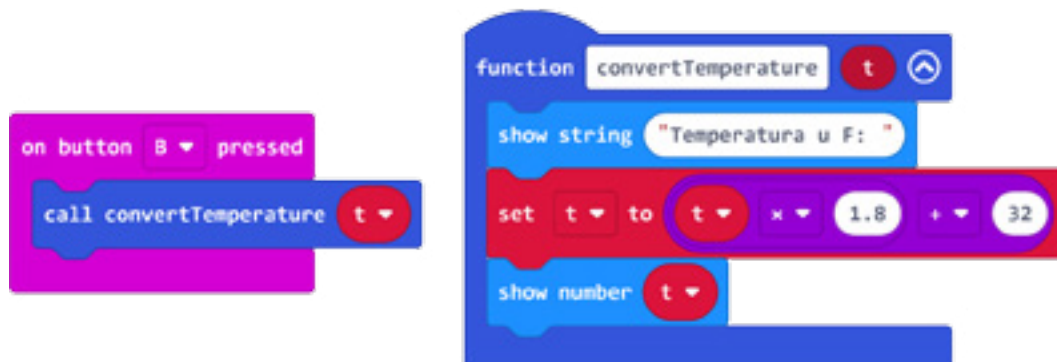
Aktivnost 6

Kreiranje *convertTemperature* funkcije – Učenici će zatim kreirati novu funkciju koja će konvertovati temperaturu izraženu u celzijusima u farenhajte. Da bi to bilo moguće, nastavnik ih upoznaje sa funkcijama koje imaju ulazne (input) parametre; u ovom slučaju, funkcija će imati parametar promjenljivu „t”. Funkciju ćemo nazvati *convertTemperature*. Klikom na *Number* dodaće nam se parametar „num”, koji treba preimenovati u „t”.



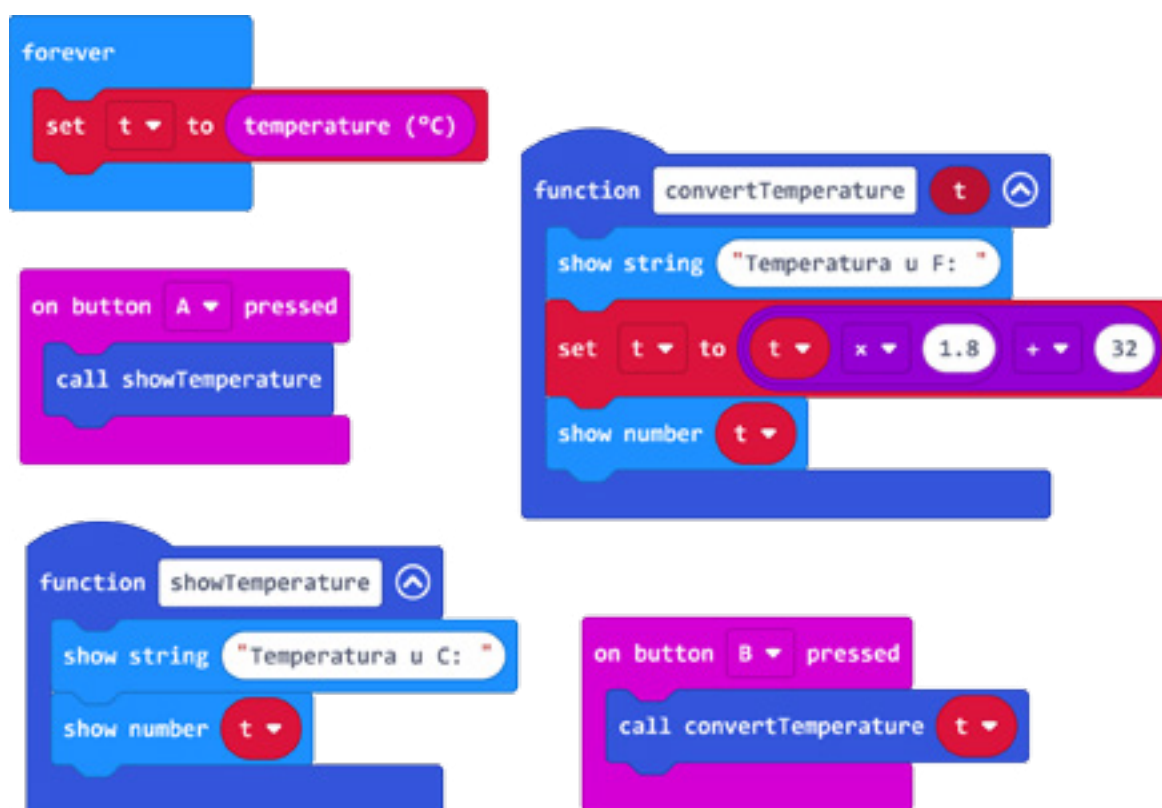
Aktivnost 7

Pozivanja funkcije – Poziv funkcije *call convertTemperature* dodaćemo u blok za klik na dugme B, i na mjestu input parametra dodatu promjenljivu „t”. U samoj funkciji iskoristićemo formulu za konvertovanje temperature.



Aktivnost 8

Finalna aktivnost – Učenici verifikuju da im je kôd funkcionalan, a finalno izgleda kao na slici.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_6hfLJrAJYiHs



PLAN ČASA BROJ 2	
OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik razumije slučajne brojeve, opšti oblik funkcije, parametre funkcija i povratnu vrijednost.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.

KRATAK OPIS ČASA
<p>Učenici će se upoznati s matematičkim zadatkom poznatim kao „$3n + 1$” problem” ili Kolacova pretpostavka (Collatz conjecture), za čiju provjeru možemo napisati program, koristeći funkcije sa argumentima. Kolacova pretpostavka kaže da ukoliko uzmemo bilo koji pozitivan cijeli broj, primjenom operacije $3n + 1$ ukoliko je broj neparan ili $n/2$ ukoliko je broj paran, sekvenca će završiti u 4,2,1 petlji.</p> <p>Algoritam provjere Kolacove pretpostavke je sljedeći:</p> <ol style="list-style-type: none"> 1. Odabrati proizvoljan pozitivan cijeli broj (u programu ćemo koristiti raspon od 1 do 100). 2. Provjeriti je li broj n paran ili neparan: <ol style="list-style-type: none"> a. paran broj: pozvati funkciju koja izračunava $n/2$, b. neparan broj: pozvati funkciju koja izračunava $3n + 1$. 3. Nakon što je rezultat sekvence dostigao vrijednost 4, program će u neprekidnom ciklusu ispisivati vrijednosti 4,2,1, čime smo provjerili Kolacovu pretpostavku za dati broj n.

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa problemom „ $3n + 1$ ”, te načinom na koji se isti može provjeriti pomoću programa koji koristi promjenljive, cikluse, predefinisane funkcije i novokreirane funkcije sa argumentima.

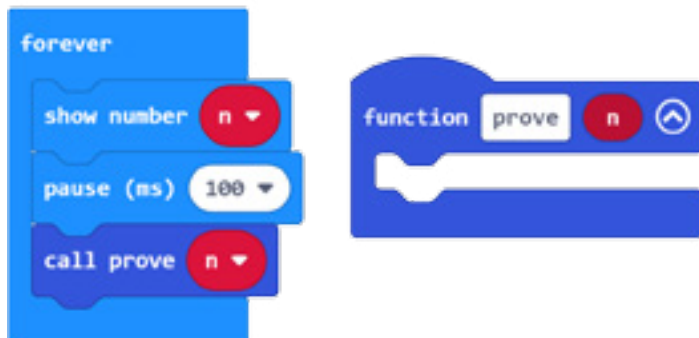
Aktivnost 2

Kreiranje promjenljivih – Na samom početku, učenik kreira promjenljivu „ n ” koja će koristiti kao ulaz (argument) u novokreiranim funkcijama, dok se njena vrijednost postavlja koristeći predefinisanu funkciju za odabir slučajnih brojeva u rasponu od 1 do 100. Na taj način će ih nastavnik upoznati sa predefinisanim funkcijama.



Aktivnost 3

Kreiranje forever block-a – S obzirom na to da Kolacova pretpostavka kaže da će sve sekvence eventualno završiti u 4,2,1 ciklusu, za provjeru će se koristiti *forever* petlja u kojoj će se prikazati vrijednost promjenljive „n“.



Aktivnost 4

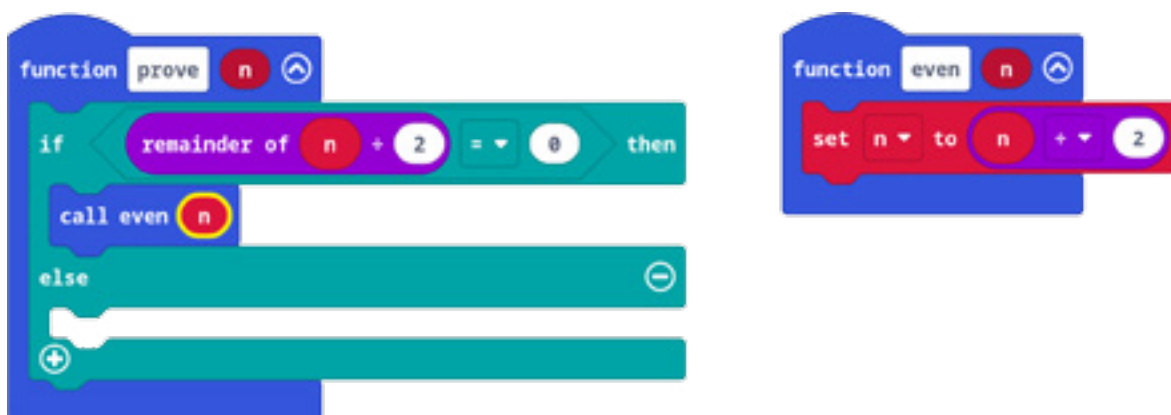
Kreiranje i poziv funkcije – Kako su učenici na prethodnom času savladali način na koji se kreiraju funkcije, tokom ove vježbe učenik kreira funkciju *prove(int n)*, koja za argument uzima „n” promjenljivu i poziva je unutar *forever* petlje.

Aktivnost 5

Kreiranje if-else block-a – Unutar funkcije *prove* provjeriće se da li je random broj paran ili neparan koristeći uslove *if-else*, kao i predefinisanu matematičku funkciju *remainder of*, koja će učeniku pokazati da li broj nakon dijeljenja sa 2 daje ostatak; ukoliko nema ostatka, broj je paran (*if* uslov je ispunjen), ili u slučaju da je ostatak 1, učenik zaključuje da je broj neparan (*else* blok).

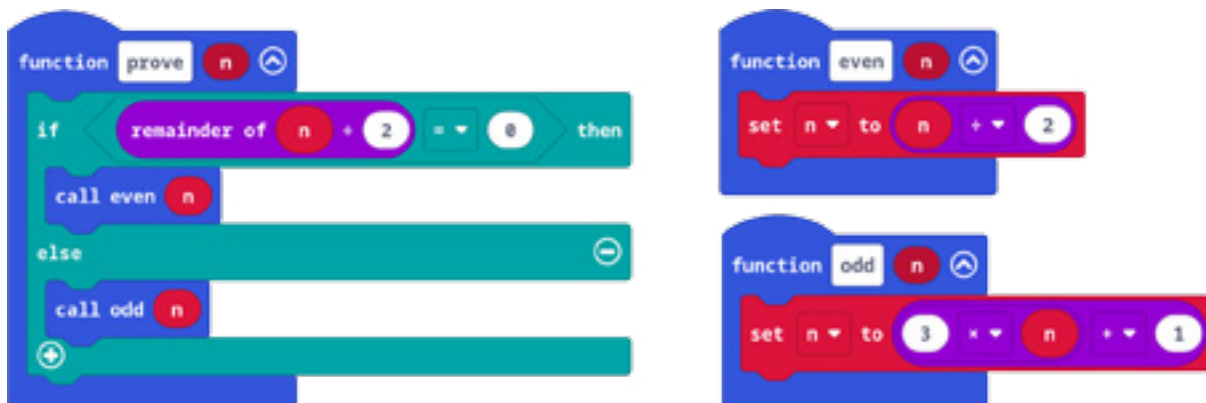
Aktivnost 6

Kreiranje tijela even funkcije – Kako Kolacova pretpostavka nalaže da se broj dijeli sa 2 ukoliko je paran, u tom slučaju u *if* bloku učenik kreira i poziva novu funkciju *even*, koja za argument takođe uzima „n” broj i dijeli ga sa 2.



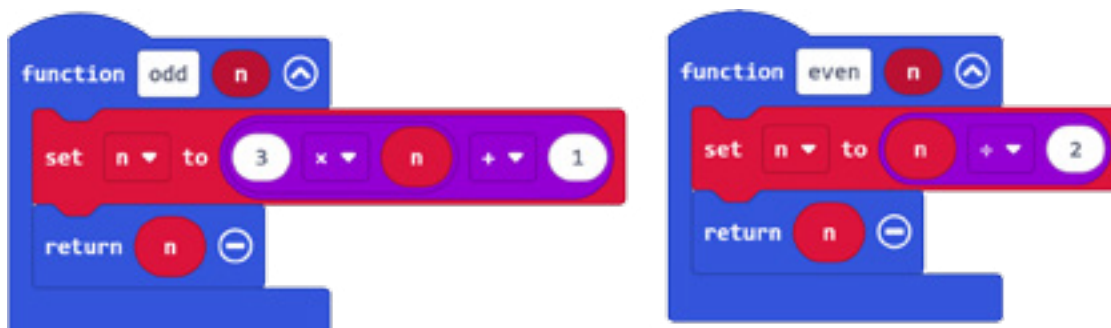
Aktivnost 7

Kreiranje tijela *odd* funkcije – Kako Kolacova pretpostavka nalaže da se broj „n“ množi sa 3 i poveća za 1 ukoliko je neparan, u tom slučaju u *else* bloku učenik kreira i poziva novu funkciju *odd*, koja za argument takođe uzima broj „n“, i obrađuje ga kako je to gore navedeno.



Aktivnost 8

Return izraz – Nakon što je učenik pokrenuo kôd, prepoznaje da nedostaju povratne informacije o tome koja je vrijednost „n“ broja nakon što se izvršila funkcija. Ovdje se učenik upoznaje sa konceptom „return“ u funkcijama, koji ima zadatak da, osim što će izvršiti neku operaciju nad nekim brojem, treba i da vrati novu vrijednost istog. Na isti način se dodaje i „return“ izraz u funkciju *prove*.



Aktivnost 9

Pozivanje napisane funkcije – Kako pozivanje funkcije vraća novu vrijednost broja „n“, potrebno je da učenik pozove „set n“ i kao parametar mu preda funkciju *prove*. Učenik se ovdje susreće s inicijalizacijom promjenljive pozivanjem funkcija.

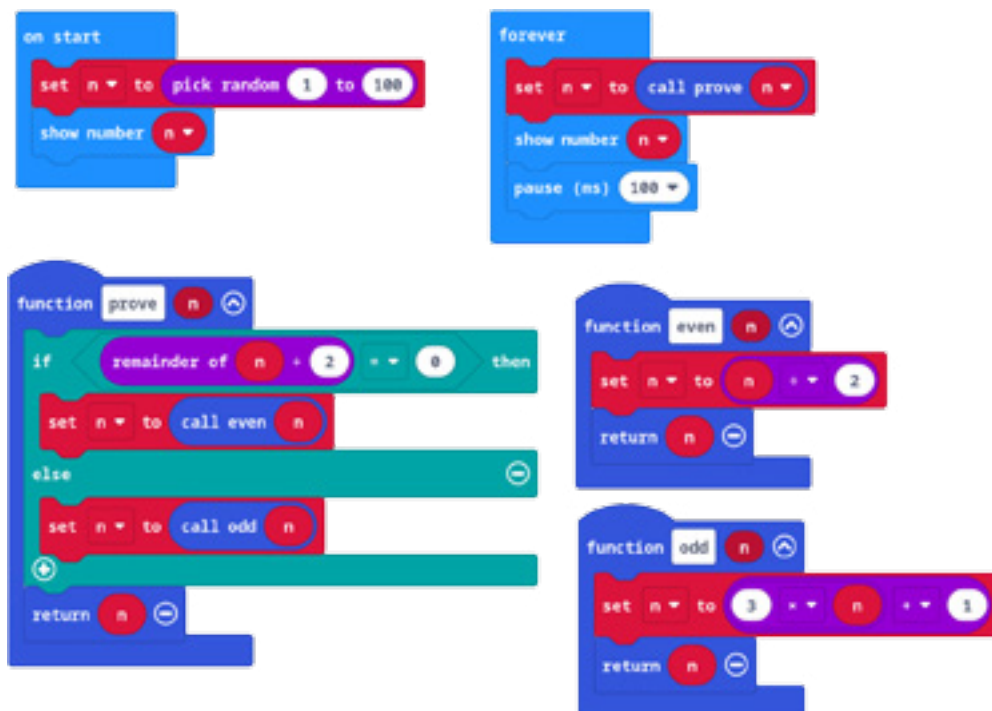
Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_MUMijVhtEDxe



Aktivnost 10

Finalna aktivnost – U završnici, program izgleda kao na slici niže, a nakon izvjesnog vremena, krenuće da prikazuje vrijednosti 4,2,1, čime su učenici provjerili Kolacovu pretpostavku za broj n.



PLAN ČASA BROJ 3

OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik razumije i koristi grafički i tekstualni način zadavanja programa.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.

KRATAK OPIS ČASA

Učenici će se upoznati sa sintaksom JavaScript-a, funkcionalnostima za kreiranje funkcija i njihovim pozivanjem unutar nekog bloka. Za potrebe učenja, kreiraćemo aplikaciju koja simulira osnovne matematičke operacije.

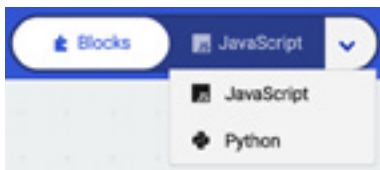
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa terminom programski jezik, upoznaje ih sa jezicima JavaScript i Python i terminom „sintaksa programskog jezika”.

Aktivnost 2

Prikaz JavaScript/Python prozora - Na samom početku učenik će kreirati novi projekat i obrisati defaultne komponente, te će ih upoznati s prozorom za JavaScript i Python kôd.



Aktivnost 3

Kreiranje promjenljivih – S obzirom na to da su za izvršenje neke matematičke operacije potrebne 2 promjenljive, učenik kreira promjenljivu „a” i promjenljivu „b”. Upoznaje sa sa ključnom riječi za definisanje promjenljivih u JavaScript-u, „let”, kao i sa korišćenjem „=” za dodjeljivanje vrijednosti. Primijetiće da se kreirao *code block* u Blocks prozoru, što je ekvivalentno napisanom kôdu:

```
let b = 0  
let a = 0
```

Aktivnost 4

Kreiranje funkcije – Nastavnik upoznaje učenika s ključnom riječju „function” za kreiranje funkcija, gdje će učenik kreirati funkciju *compare* kako bi uporedio vrijednosti promjenljive „a” i promjenljive „b”. Nastavnik upoznaje učenike sa upotrebom „()” nakon imena funkcija, koje se koriste da bi se funkcijama dodijelili ulazni (input) parametri, a ukoliko je funkcija bez parametara, zagrada je prazna. „{}” se koriste kako bi se ogradio tijelo jedne funkcije od ostatka kôda:

```
function compare () {  
}
```

Aktivnost 5

Kreiranje tijela funkcije – Učenik će u tijelu funkcije, koristeći logički *if-else*, uporediti vrijednosti promjenljivih „a” i „b”. Ključne riječi za ovaj zadatak su kao gorepomenute, osim što se tijelo takođe ograđuje „{}”. Zagrade „()” se u ovom slučaju koriste za ispisivanje uslova koji provjeravamo, u ovom slučaju da li je „a” manje ili veće od „b”, dok nam je *else* blok rezervisan za izlaz (output) ukoliko su „a” i „b” jednakih vrijednosti.

Aktivnost 6

Kreiranja tijela funkcije, nastavak – Ukoliko je bilo koji od ovih uslova ispunjen, učenik piše kôd za prikaz odgovarajućeg izraza, koristeći integrisane funkcije MakeCode-a, koje uglavnom počinju sa ključnom riječju „basic”, koja je za prikaz nekog izraza „showString”:

```
function compare () {  
    if (a < b) {  
        basic.showString("a je manje od b")  
    } else if (a > b) {  
        basic.showString("a je veće od b")  
    } else {  
        basic.showString("a i b su jednakih vrijednosti")  
    }  
}
```

Aktivnost 7

Poziv funkcije – Učenik prepoznaje da je funkcija kreirana, ali da nije nigdje pozvana. Ideja je da se ista pozove klikom na dugme A, za šta takođe koristimo integrisanje input operacije *MakeCode*-a, *onButtonPressed*. Unutar tijela ove funkcije za poziv funkcije *compare* unijećemo ime funkcije i „()“, preko kojih učenik ubuduće prepoznaje pozive funkcija:

```
input.onButtonPressed(Button.A, function () {  
    compare()  
})
```

Aktivnost 8

Kreiranje funkcije s argumentima – Učenik se u prethodnom dijelu upoznao sa funkcijama bez argumenata, a sada će kreirati funkciju koja će za input parametre imati promjenljive „a“ i „b“, sabrati ih i njihov zbir čuvati u novoj promjenljivoj. Funkciju ćemo nazvati *add*. Koristeći ključnu riječ „number“, učenik uči program da prepozna da je input parameter tipa broj, što mora biti s obzirom na to da će izvršiti sabiranje. Kako bismo sačuvali vrijednost zbira ova dva broja, kreiraćemo novu promjenljivu „sum“:

```
function add (a: number, b: number) {  
    sum = a + b  
}
```

Aktivnost 9

Upotreba return izraza – S obzirom na to da imamo novu promjenljivu „sum“ koja se inicijalizuje unutar tijela funkcije *add*, učenik se upoznaje s ključnom riječju „return“, koja se koristi za funkcije koje imaju neku povratnu vrijednost:

```
function add (a: number, b: number) {  
    sum = a + b  
    return sum  
}
```

Aktivnost 10

Poziv funkcije na dugme B – Pozivom ove funkcije učenik će ponovo kreirati input operaciju *onButtonPressed*, ali ovoga puta dugmeta B. Nastavnik upoznaje učenike sa integrisanom funkcijom *showNumber*, koja kao parametar može uzeti poziv *add* funkcije, čime se dokazuje da je povratna vrijednost funkcije *add* zaista neki broj, tj. zbir brojeva „a“ i „b“:

```
input.onButtonPressed(Button.B, function () {  
    basic.showNumber(add(a, b))  
})
```

Aktivnost 11

Finalna aktivnost – Učenik testira kôd koji finalno izgleda:

```
// Poziv funkcije compare na klik dugmeta A
input.onButtonPressed(Button.A, function () {
    compare()
})
function compare () {
    if (a < b) {
        basic.showString("a je manje od b")
    } else if (a > b) {
        basic.showString("a je veće od b")
    } else {
        basic.showString("a i b su jednakih vrijednosti")
    }
}

// Poziv funkcije add na klik dugmeta B
input.onButtonPressed(Button.B, function () {
    basic.showNumber(add(a, b))
})
function add (a: number, b: number) {
    sum = a + b
    return sum
}

//Promjenljive
let sum = 0
let b = 0
let a = 0
a = 5
b = 10
```

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_9bKiWmfCY8Vk



PLAN ČASA BROJ 4

OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik kreira programe koji koriste senzore.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, servo motor (Mini 180 Degree Resin Gear Servo SG90), taster (Keyestudio Digital Push Button), 6 x Ž/Ž "jumper" žice, 6 x krokodilke i Micro:bit, kao i set baterija 2 x 1,5 V.

KRATAK OPIS ČASA

Učenici će samostalno, čitajući šemu, povezati servo motor na Micro:bit i napisati kôd za njegovu kontrolu pomoću dugmeta A, za otvaranje, odnosno rotaciju od 90 stepeni do 0, ili klikom na dugme B, zatvaranje, odnosno rotaciju od 90 stepeni do 180. Koristeći eksterni taster, pokrenućemo servo motor u kontinuiranom modu. Na ovaj način se učenik upoznao sa načinom mogućih kretanja servo motora.

FS90 je analogni servo iz FEETECH-a (ranije poznat kao Fitec), koji je po obliku i funkciji sličan Toverpro SG90. Specifikacije navode da ovaj servo ima radni ugao od 180° za standardne servo impulse između 900 µs i 2100 µs. Kao i kod većine servo uređaja, opseg impulsa se može proširiti kako bi se postigao prošireni radni ugao.

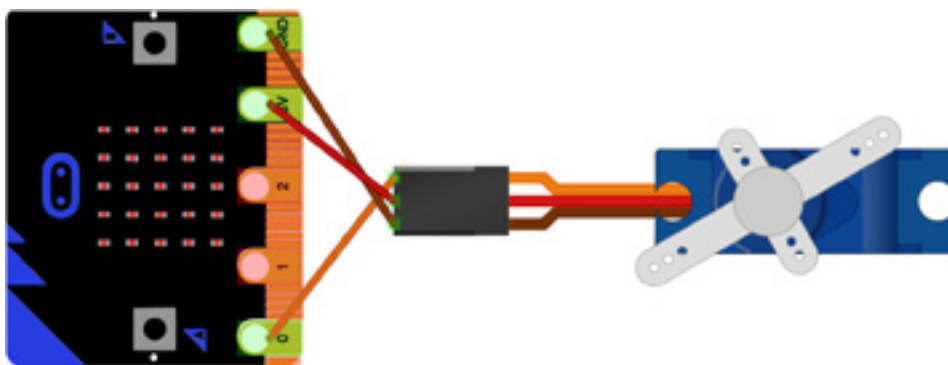
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Povezivanje komponenti – Učenici uzimaju set opreme neophodne za kreiranje šeme koju im je nastavnik prikazao. Nastavnik objašnjava kodiranje šema boja, crna/braon za uzemljenje (GND = ground), crvena za izvor napajanja (3V pin) i druge boje za kontrolne pinove.

Aktivnost 2

Povezivanje šeme – Koristeći krokodilke i "jumper" žice, učenici će povezati braon izlaz servo motora na GND pin, crveni izlaz na 3V, a za kontrolu povezaće žuti izlaz na P0. Na glavu servo motora učenik će dodati jednokrilni dodatak radi lakšeg pregleda rotacija servo motora. Šema za povezivanje servo motora na Micro:bit je kao na slici.



Izvor: [microbit.org](https://support.microbit.org)³

³ <https://support.microbit.org/support/solutions/articles/19000101864-using-a-servo-with-the-micro-bit>

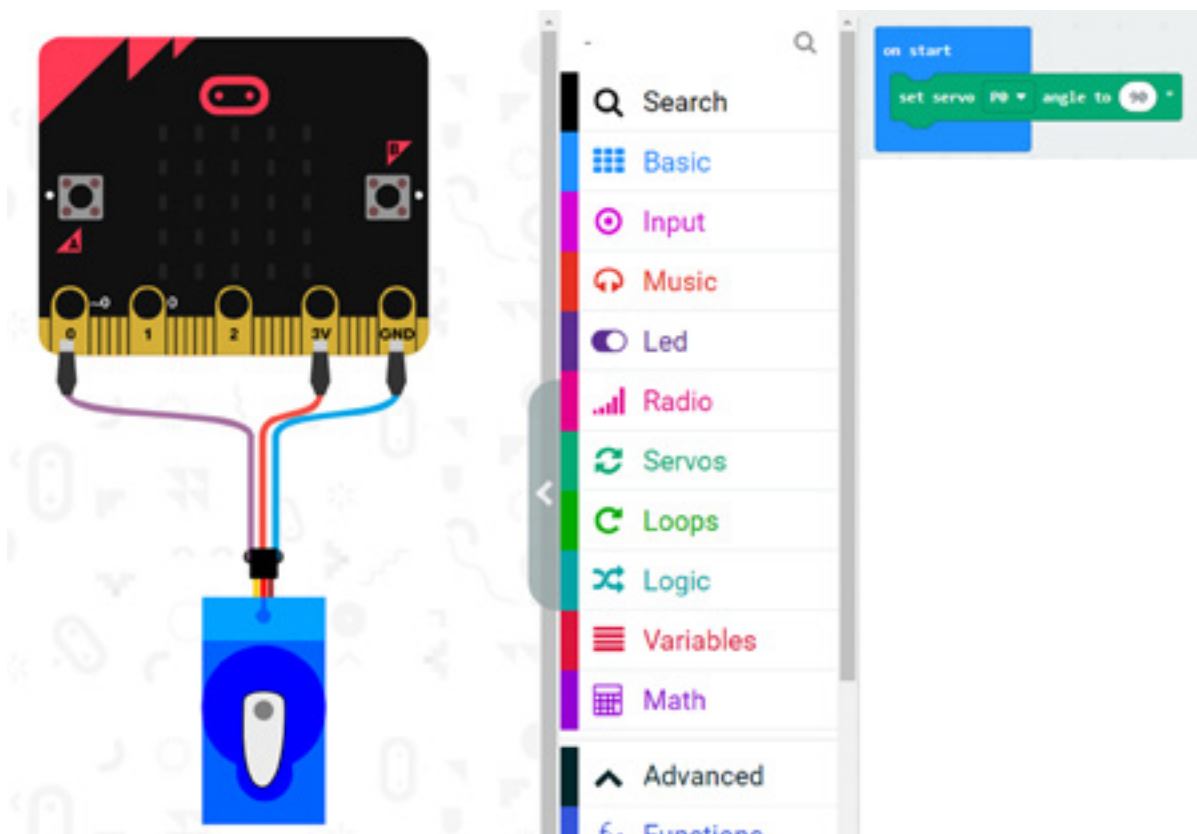
Aktivnost 3

Upoznavanje s ekstenzijom za servo motor – Učenik zatim otvara MakeCode okruženje i dodaje ekstenziju za servo motor. To će uraditi tako što će mu se klikom na *Advanced* -> *Extensions* otvoriti prozor sa ekstenzijama, u kome će naći ekstenziju za servo, pri čemu će se kreirati novi panel *Servos* s dodatnim komandama.



Aktivnost 4

Postavljanje startne pozicije – Kako bi učenik postavio početni položaj servo motora na 90 stepeni, potrebno je da prevuče set metodu u *onStart* blok. Primijetiće da je simulator već na startu programa dodao servo komponente i postavio vrijednost servo motora na 90 (paralelno s dužom stranom tijela servo motora).



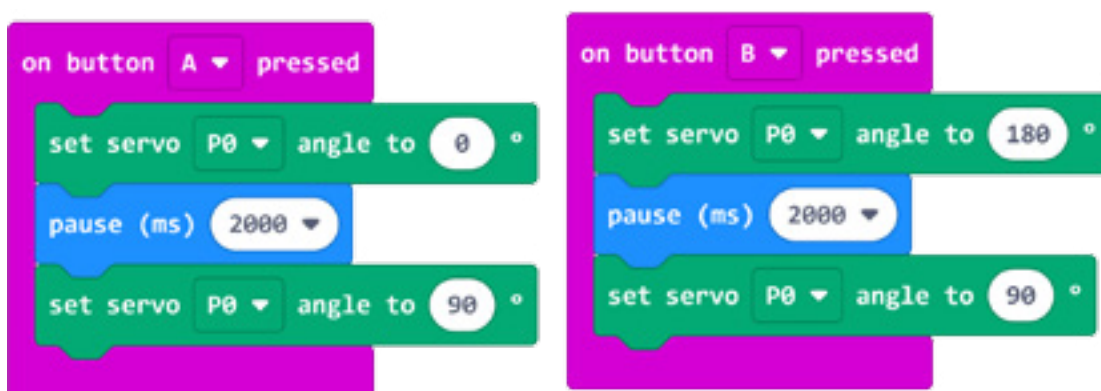
Aktivnost 5

Kreiranje tijela funkcije – Učenik će zatim dodati kontrole za klik na dugme A i dugme B.



Aktivnost 6

Kreiranja tijela funkcije, nastavak – Kako je na početku definisan zadatak da se klikom na dugme A otvara (0 stepeni), a klikom na dugme B zatvara (180 stepeni), učenik će u ovim funkcijama ponovo pozvati set metodu, te će nakon 2 sekunde ponovo isti zatvoriti. Nastavnik zatim objašnjava da se servo motor može koristiti za ostvarivanje bilo kakvog ograničenog ugaonog kretanja u rasponu od 0 stepeni do 180 stepeni, što se može prikazati na simulatoru.



Aktivnost 7

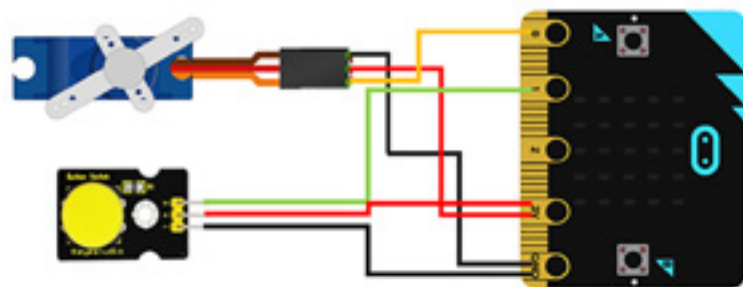
Inicijalno testiranje šeme – Učenik zatim preuzima kôd kao .hex fajl i prebacuje ga na Micro:bit uređaj, na koji je prethodno povezao servo motor, preko USB kabla. Učenik primjećuje da je napajanje Micro:bit-a preko USB-a nestabilno, pa će na ulaz za napon povezati set od 2 x 1,5 V baterija sa kućištem, što će omogućiti stabilno napajanje Micro:bit-a i servo motora.

Aktivnost 8

Dodavanje eksternog tastera u šemu – Učenik uočava da se servo motor može pokretati pomoću integriranih dugmića, gdje kao naredni zadatak treba da poveže eksterno dugme ili taster (koje se takođe može naći u setu senzora), koje će omogućiti kružno kretanje servo motora po procentu maksimalne brzine (50%).



Povezivanje će izvršiti na isti način kao i za servo motor, osim što će kontrolni pin povezati na P1, kao na slici:



Aktivnost 9

Testiranje kontinuiranog kretanja servo motora – Učenik zatim dodaje funkcionalnost iz basic elemenata, koja se zove *on pin pressed*, što će mu omogućiti da primi informaciju da li je dugme pritisnuto. Unutar samog tijela ove funkcije učenik dodaje komponentu za omogućavanje kontinuiranog kružnog kretanja servo motora, te to izgleda ovako:



Napomena: Nemaju svi servo motori mogućnost kontinuiranog kružnog kretanja. U setu dodatne Micro:bit opreme postoji servo motor ovog tipa.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_05yX3U6uCARR



PLAN ČASA BROJ 5	
OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik kreira programe koji koriste senzore.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, DC motor (Motors High Torque), minijaturni 51mm četvorokrilni bijeli propeler (51mm White Four Blade Miniature Impeller), relej (Single Relay Module), "jumper" žice, 4 x krokodilke i Micro:bit, kao i set baterija 2 x 1,5 V.
KRATAK OPIS ČASA	
Na ovom času učenik se upoznaje s drugim tipom motora, tzv. DC motorom. Osnovna razlika između DC motora i servo motora je ta što DC motor ima kontinuirani obrtni momenat, dok servo motor posjeduje mogućnost držanja trenutne pozicije. Koristeći integrisani termometar, provjerićemo trenutnu temperaturu, gdje bi se uslovno uključivao, odnosno isključivao ventilacioni sistem, koji će biti simuliran DC motorom na koji je povezan S propeler.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Upoznavanje s principom rada DC motora – Nastavnik upoznaje učenike s principom rada DC motora, odnosno motora s malim obrtnim momentom, koji za razliku od servo motora ima kontinuirani obrt bez mogućnosti ugaone kontrole. Nominalni napon za rad motora je 3,0 V, s operativnim rasponom od 1,4 do 4,5 V.

Aktivnost 2

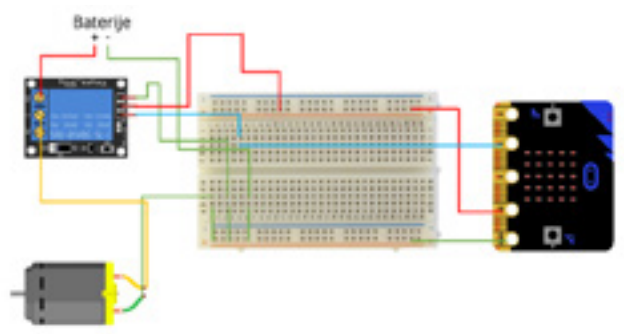
Priprema servo motora – Učenici će na svom servo motoru povezati minijaturni četvorokrilni bijeli propeler kako bi simulirali ventilacioni uređaj koji radi na sličnom principu.

Aktivnost 3

Upoznavanje s radom releja – Nastavnik upoznaje učenike sa tim da je za povezivanje motora potreban relej, odnosno elektronska komponenta, koja ima ulogu prekidača, odnosno sistema koji koristi elektromagnete za prekidanje/uspostavljanje protoka struje u strujnom kolu.

Aktivnost 4

Povezivanje komponenti pomoću šeme – Učenik, prateći šemu, kreira strujno kolo za uspostavljanje sistema za ventilaciju, a šema izgleda kao na slici:



Aktivnost 5

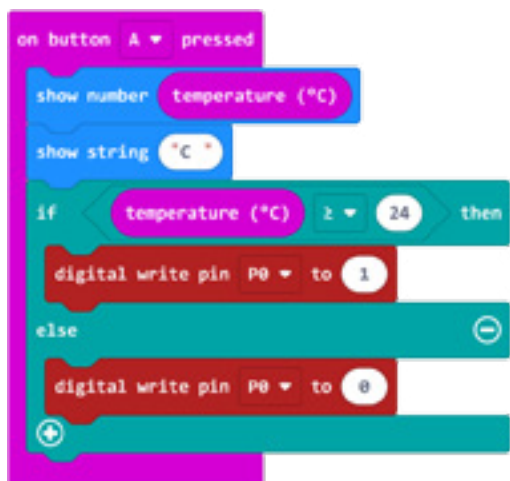
Kreiranje promjenljivih – Učenik otvara MakeCode okruženje kako bi napisao kôd za kontrolu sistema. S obzirom na to da je ideja da se ventilacija pokrene klikom na interno dugme, i to kada temperatura bude iznad 24 stepena, potrebna nam je promjenljiva koja će sačuvati vrijednost temperature na klik dugmeta A.

Aktivnost 6

Kreiranja tijela funkcije – Učenik zatim dodaje logiku za uslovno paljenje ventilacije, odnosno DC motora. Učenik kreira *if-else* logički blok i postavlja uslov u kome provjeravamo da li je temperatura iznad optimalna 24 stepena.

Aktivnost 7

Kreiranja tijela funkcije, nastavak – Uzimajući u obzir da je šematski relej, odnosno prekidač, zadužen za pokretanje sistema postavljen na kontrolni pin 1, učenik dodaje *digital write* komandu za pokretanje obrtnog kretanja motora, odnosno postavlja vrijednost na 1 (relej uspostavlja strujno kolo), a u *else* slučaju postavlja vrijednost na 0, gdje relej ponovo prekida strujno kolo:



Aktivnost 8

Kreiranje kôda za zaustavljanje sistema – Ukoliko postoji potreba za manuelnim gašenjem sistema za ventilaciju, učenik će dodati komandu za gašenje sistema za ventilaciju na klik dugmeta B.

Aktivnost 9

Finalna aktivnost – Učenik je upoznat sa tim da je nominalni napon za rad DC motora 3,0 V, gdje je dovoljno da se nakon prebacivanja kôda na Micro:bit sistem poveže na set od 2 x 1,5 V baterije.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_TdjY6xXryHrg



PLAN ČASA BROJ 6	
OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik kreira programe koji koriste senzore.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, UV senzor (GUVA-S12SD 3528 Ultraviolet Sensor), 3 x M/M “jumper” žice, 6 x Ž/Ž “jumper” žice, 6 x krokodilke i Micro:bit, kao i set baterija 2 x 1,5 V.
KRATAK OPIS ČASA	
Učenici će samostalno, čitajući šemu, povezati ultraljubičasti senzor (GUVA-S12SD 3528 Ultraviolet Sensor) i pinove na Micro:bit uređaj i napisati kôd za njegovu kontrolu. Ideja je da ukoliko senzor registruje veći nivo UV svjetlosti, učenik će napraviti kôd za pokretanje servo motora koji bi trebalo da simulira otvaranje suncobrana, gdje se inače ljudi najviše izlažu UV svjetlosti Sunca.	

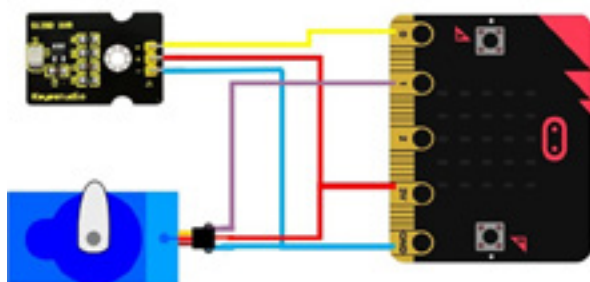
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Upoznavanje s principom rada UV senzora – Nastavnik upoznaje učenike sa UV senzorom, te sa tipom ultraviolettne svjetlosti iz spektra nevidljive svjetlosti, kao i da se ona emituje svakodnevno. Zatim ih upoznaje sa senzorom koji može da registruje nivoe ove svjetlosti.

Aktivnost 2

Povezivanje komponenti pomoću šeme – Čitajući šemu, učenik povezuje UV senzor na Micro:bit. Pin 1 senzora je GND na Micro:bit-u, pin 2 senzora je napajanje, dok je pin 3 senzora kontrolni pin i povezuje se na P0 Micro:bit-a.



Aktivnost 3

Kreiranje promjenljive – Učenik će zatim kreirati promjenljivu UV, čija vrijednost će se dodjeljivati čitanjem analognog signala sa P0 pina, na koji je povezan senzor.

Aktivnost 4

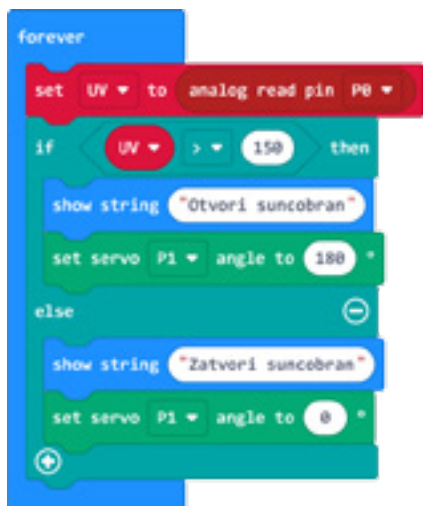
Postavljanje referentne vrijednosti – S obzirom na to da su vrijednosti analognog signala u rasponu od 0 do 1024, uzećemo da je UV svjetlost sa analognom vrijednošću većom od 150 štetna i da se tada suncobran mora otvoriti.

Aktivnost 5

Dodavanje servo motora u šemu – Pošto je testirao da kôd zaista čita vrijednosti sa UV senzora, učenik će u šemu dodati servo motor, kako je to ranije objašnjeno, a kontrolni pin će povezati na pin P1 Micro:bit uređaja.

Aktivnost 6

Kreiranje tijela forever block-a – Rekli smo da nam je simulator za suncobran rotacija servo motora, gdje ćemo mu kao početno stanje (zatvoreno) postaviti 0 stepeni, a stanje otvoreno 180 stepeni. Kako nam je UV senzor povezan na P0, kontrolni pin za servo motor postavimo na P1.



Aktivnost 7

Finalna aktivnost – Učenik zatim preuzima kôd i prebacuje ga na Micro:bit, te povezivanjem eksternog napajanja testira funkcionalnost sistema.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_4XHHLED4xLaV



PLAN ČASA BROJ 7	
OVI 5	Na kraju učenja učenik će biti osposobljen da implementira funkcije i programe koji koriste senzore.
Ishodi učenja	Na kraju učenja učenik kreira programe koji koriste senzore.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, senzor za mjerenje vlažnosti zemljišta (Soil Humidity Sensor), proto-ploča (breadboard) 3 x M/M "jumper" žice, 3 x Ž/Ž "jumper" žice, 3 x krokodilke i Micro:bit, kao i set baterija 2 x 1,5 V.
KRATAK OPIS ČASA	
Učenici će samostalno, čitajući šemu sa prototipskom pločom, povezati senzor za vlažnost i pinove na Micro:bit uređaj i napisati kôd za njegovu kontrolu. Učenik će napisati kôd koji prikazuje „srećni smiješak“ ukoliko je vlažnost na zadovoljavajućem nivou, a „tužni smiješak“ ukoliko je vlažnost niža od zadovoljavajuće.	

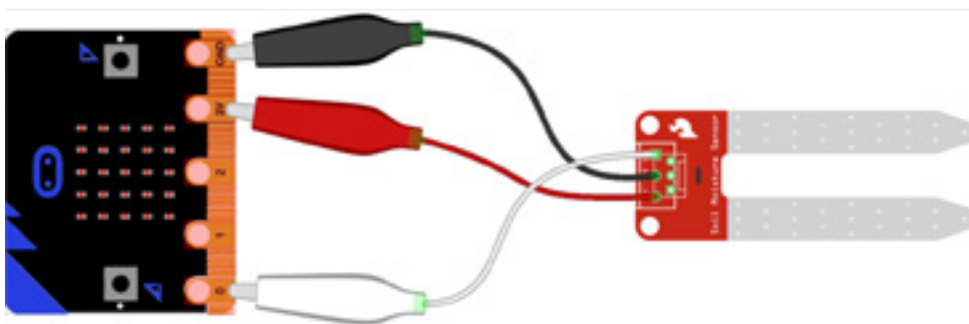
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Upoznavanje s proto-pločom – Nastavnik upoznaje učenike s proto-pločom za povezivanje komponenti, gdje među horizontalnim rupicama postoji veza, s prekidom na sredini, dok su vertikalne bočne rupice povezane po vertikali.

Aktivnost 2

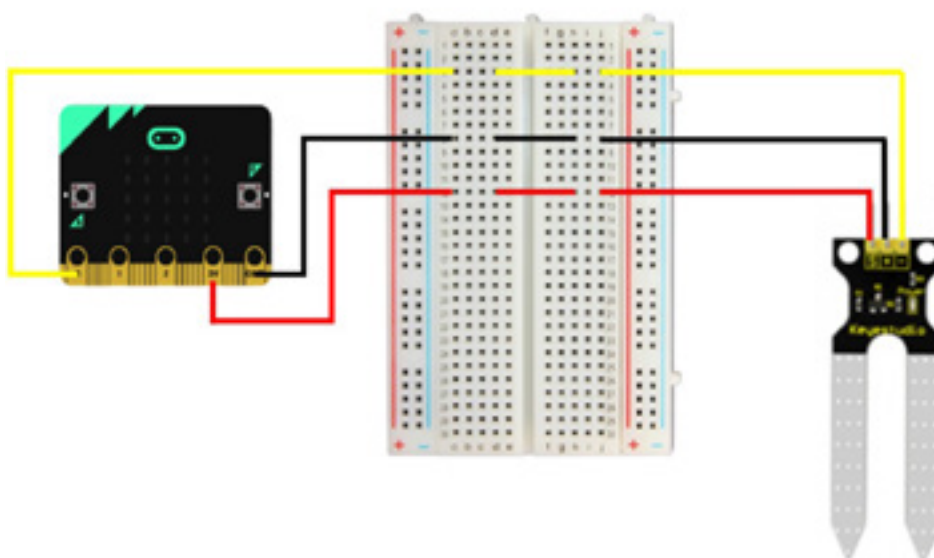
Povezivanje senzora bez proto-ploče – Učenik će najprije samostalno povezati senzor za vlažnost zemljišta koristeći krokodilke i džumpere (koristeći šemu).



Izvor: core-electronics.com.au⁴

Aktivnost 3

Povezivanje senzora s proto-pločom – Istu šemu nastavnik će pokazati pomoću proto-ploče, a koja izgleda ovako:



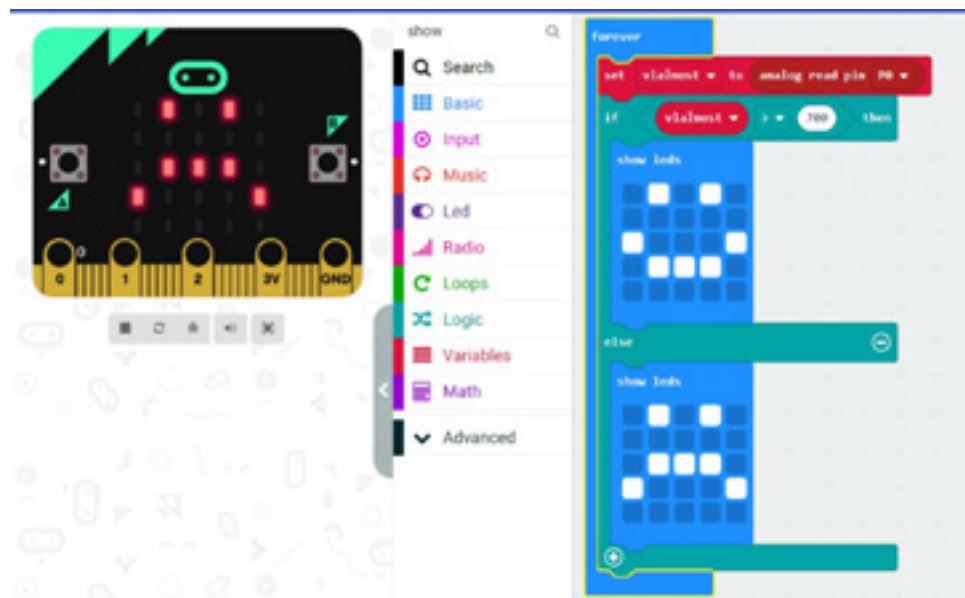
Aktivnost 4

Kreiranje i inicijalizacija promjenljive – Nakon što je učenik uspješno povezao senzor preko prototipske ploče, otvara MakeCode okruženje i piše kôd za njegovu kontrolu. Prvo će kreirati promjenljivu čiju će vrijednost zadati čitajući analogni pin P0.

⁴ <https://core-electronics.com.au/tutorials/microbit-soil-moisture-sensor.html>

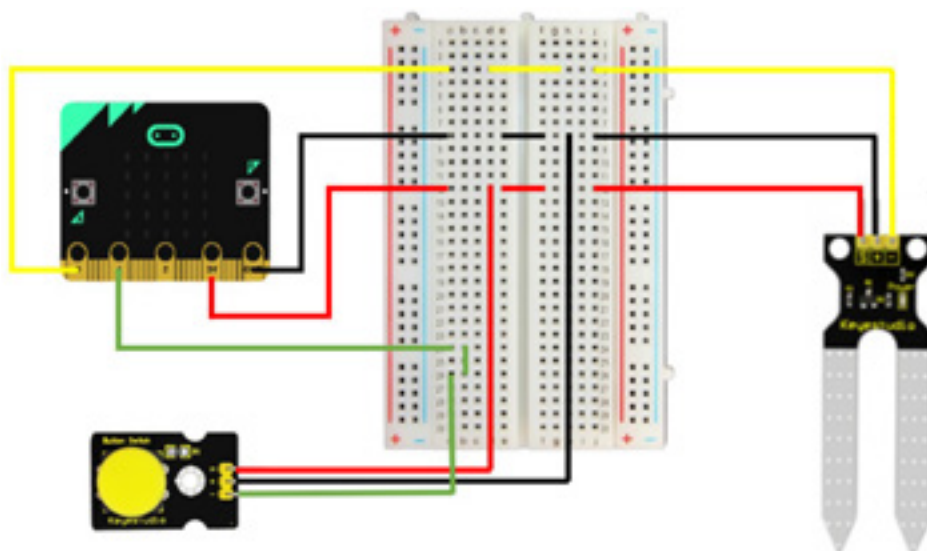
Aktivnost 5

Kreiranje tijela forever block-a – Učenik će zatim napisati kôd da prikaže „srećni smiješak“ ukoliko je vlažnost na zadovoljavajućem nivou, pretpostavimo da je to 700, a „tužni smiješak“ u drugim slučajevima.



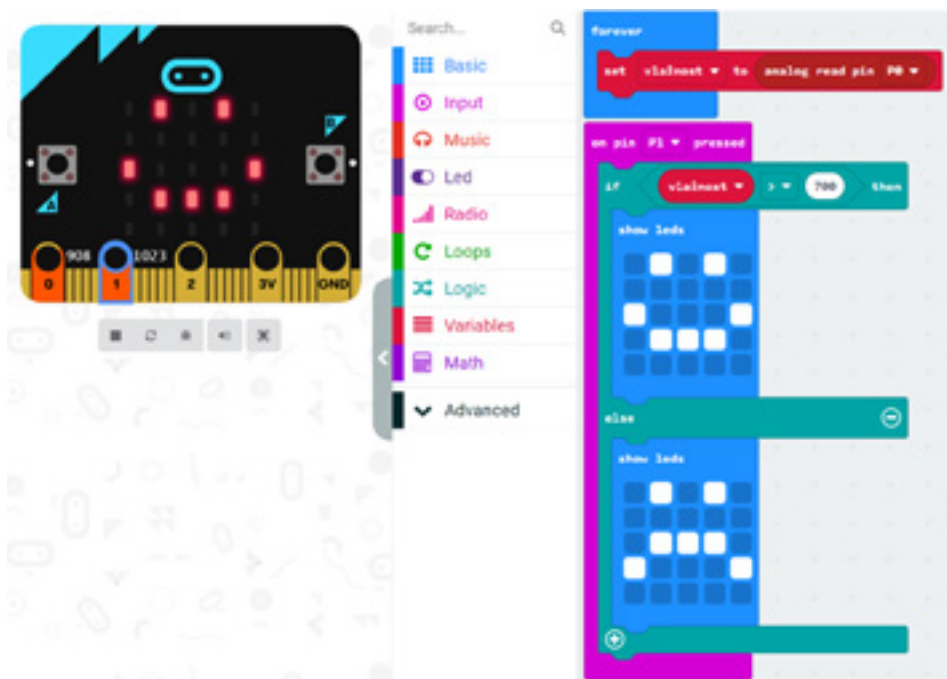
Aktivnost 6

Dodavanje eksternog tastera u šemu – Nastavnik upućuje učenike u naredni zadatak, gdje ćemo prikaz vlažnosti inicijalizovati tek na klik eksternog tastera. Učenici će samostalno dodati isti u svoje strujno kolo prema zadatoj šemi:



Aktivnost 7

Adaptacija kôda – Učenici će zatim adaptirati kôd, gdje će se, umjesto u *forever* petlji, prikaz smajlića dešavati na klik tastera, odnosno dodaće *on pin P1 pressed*, na koji je povezan isti.



Aktivnost 8

Finalna aktivnost – Učenik će zatim preuzeti kôd, unijeti ga u Micro:bit, povezati ga na set baterija za napajanje i testirati funkcionalnost i ispravnost šeme.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_R39iHshVRdsc





PRIMJERI PRAĆENJA PROCJENJIVANJA

Akcenat obrazovno-vaspitnih ishoda koji su obuhvaćeni ovim priručnikom je na praktičnom znanju iz oblasti programiranja i elektronike, pa je suština aktivnosti u fazi provjere i ocjenjivanja znanja na radu sa praktičnim vježbama, a ne samo na teorijskim znanjima iz oblasti ICT-a. Brojčanom ocjenom (od 1 do 5) treba vrednovati praktične aktivnosti, ali i pokazano znanje i razumijevanje. Takođe, treba vrednovati kreativnost, originalnost rješenja, radno-tehničku spretnost, uvažavajući interesovanja, zalaganja i individualne sposobnosti učenika.

Primjeri kriterijuma ocjenjivanja:

VII RAZRED		
NIVO POSTIGNUĆA		
Osnovni nivo	Srednji nivo	Viši nivo
<ul style="list-style-type: none"> - prepoznaje funkciju u MakeCode okruženju, razlikuje funkcije s parametrima i bez parametara; - prepoznaje poziv funkcije u blokovima, kao i u JavaScript-u; - objašnjava generalnu namjenu senzora; - djelimično slaže blokove za izvršenje određenog zadatka; - rijetko se uključuje u diskusiju a zadatke završava tek nakon upozorenja na moguće negativne posljedice. 	<ul style="list-style-type: none"> - samostalno ili uz pomoć nastavnika kreira funkcije sa ili bez parametara, te ih poziva u nekom bloku; - definiše pojam funkcije; - definiše pojam „sintaksa programskog jezika”, te samostalno koristi neke od ključnih riječi za pisanje kôda u JavaScript-u; - koristi i povezuje šeme sa sensorima na Micro:bit uređaj; - povremeno se uključuje u diskusiju na času, a zadatke realizuje uz određene napomene. 	<ul style="list-style-type: none"> - samostalno stvara program kao rješenje problema, koje uključuje upotrebu funkcija za povećanje čitljivosti kôda; - samostalno kreira šeme koje podrazumijevaju kombinovanje više senzora; - samostalno piše kôd u sintaksi programskog jezika JavaScript; - razumije princip rada senzora, pojedinačno; - predviđa odgovarajuće ulazne (testne) primjere, te kritički provjerava ispravnost rješenja i prema potrebi preuređuje svoje rješenje; - često postavlja pitanja i učestvuje u diskusijama; - samostalno i na vrijeme završava zadatke, digitalni radovi su uredni, precizni i razumljivi, pomaže drugima.



PROJEKтна NASTAVA

Primjeri za izradu projekata:

1. <https://www.robotique.tech/robotics/control-a-dc-motor-with-microbit/>
2. <https://www.instructables.com/Automatic-Plant-Watering-System-Using-a-Microbit/>
3. <https://support.microbit.org/support/solutions/articles/19000101864-using-a-servo-with-the-micro-bit>
4. Izvorni kôd Python projekta koji koristi Servo Driver Board: <https://github.com/KitronikLtd/micropython-microbit-kitronik-16-servo-board>
5. <https://www.youtube.com/watch?v=5iEjguxN9q4>



OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA

- Micro:bit
- Mikro USB kabl
- Servo motor (Mini 180 Degree Resin Gear Servo SG90), servo motor 360 (Mini 360 Degree Resin Gear Servo SG90)
- Minijaturni DC motor sa malim obrtnim momentom (Miniature Low Torque Flat DC Motor)
- Minijaturni 51 mm četvorokrilni bijeli propeler (51 mm White Four Blade Miniature Impeller)
- Releji (Single Relay Module)
- Taster (Keyestudio Digital Push Button)
- Baterija sa 2 x 1,5 V baterije
- Krokodil-kopče
- Ultraljubičasti senzor (GUVA-S12SD 3528 Ultraviolet Sensor)
- Senzor za mjerenje vlažnosti zemljišta (Soil Humidity Sensor)





VIII RAZRED

OBRAZOVNO-VASPITNI ISHOD 8

Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.

Ishodi učenja

Po završetku učenja učenik će moći da:

- objasni potrebu za upotrebom nizova u programima;
- kreira nizove u programu;
- koristi osnovne operacije sa nizovima;
- kreira programe koji koriste nizove za čuvanje podataka;
- demonstrira program koji povezuje uređaje;
- kreira programe koji koriste radio i povezane računare za slanje i primanje podataka;
- implementira programe koji učitavaju podatke iz fajla.

Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda

a) Sadržaji/pojmovi:

Prilikom obrade sortiranja, na crtežu ili animaciji objasniti sortiranje niza brojeva korak po korak (veza sa ishodom 2 – algoritmi za sortiranje brojeva); prilikom obrade, programi koji koriste povezane uređaje za slanje i primanje podataka, koristiti raspoložive uređaje (npr. slanje slike ili teksta kroz mrežu Micro:bit računara; Morzeova azbuka...) – veza sa ishodom 3; prilikom pisanja programa koji učitavaju podatke iz fajla i upisuju podatke u fajl koristiti npr. upisivanje podataka u CSV fajl i primijeniti program za tabelarne proračune – veza sa ishodom 2 iz VII razreda. Pojmovi: niz, element niza, indeks, sortiranje, sortiranje selekcijom, Bubble sort, radio, broadcast poruke, učitavanje iz fajla, upisivanje u fajl.

Napomena: Složeniji algoritmi sortiranja (Radix sort, Quicksort, Mergesort...) mogu se obraditi za talentovane učenike.

b) Aktivnosti učenja

Učenici:

- po grupama diskutuju o primjerima potrebe za upotrebom nizova (npr. spisak učenika u dnevniku, niz ocjena iz matematike za jedno odjeljenje...);
- na papiru ispisuju korake sortiranja niza brojeva (veza sa ishodom 2 – algoritmi);
- uvježbavaju pisanje programa u kojima kreiraju nizove i primjenjuju osnovne operacije sa nizovima, kao što su pristup elementu niza i promjena vrijednosti elementa;
- po parovima uvježbavaju implementiranje programa koji koriste povezane uređaje za slanje i primanje podataka;
- uvježbavaju pisanje programa koji učitavaju podatke iz fajla (npr. CSV fajla) i upisuju podatke u fajl (npr. vrijeme i temperaturu upisati u CSV fajl i nacrtati grafikon pomoću programa za tabelarne proračune).

c) Broj časova realizacije (okvirno) – 6 časova.

*OBRAZOVNO-VASPITNI ISHOD 8 preuzet iz Plana i programa predmeta Informatika sa tehnikom.



DIDAKTIČKO-METODIČKA UPUTSTVA

Opšte didaktičke preporuke

Akcenat ovoga predmeta, sa posebnim osvrtom na ishode koji se odnose na programiranje, stavlja se na praktično znanje, pa je suština aktivnosti u učenju bazirana na praktičnim vježbama uz asistenciju nastavnika.

U skladu s dinamikom razvoja tehnoloških procesa i izvjesnih tehničkih inovacija, planirani nastavni sadržaji ovoga predmeta biće podložni izmjenama i dopunama. Zato ovaj priručnik, u njegovoj realizaciji, ostavlja slobodu svim učesnicima procesa.

Predviđeni sadržaji zahtijevaju raznovrstan metodički pristup, tj. obuhvataju korišćenje određenog programskog jezika, upotrebu raznih uređaja, upoznavanje i upravljanje raznim komponentama – ultrazvučni senzori, relej, servo motor, senzor pokreta, DC motora, foto-otpornika i sl.

Didaktičke preporuke za realizaciju OVI

U obrazovno-vaspitnom ishodu 8 treba insistirati na povezivanju sa već realizovanim ishodima; demonstrirati algoritme sortiranja, komunikaciju između povezanih uređaja i program za upisivanje u fajl; koristiti uređaje Micro:bit ili druge slične uređaje. Od talentovanih učenika treba tražiti da kreiraju projekte koji su kombinacija različitih ishoda (npr. da naprave maketu kuće ili stana i demonstriraju kontrolu osvjjetljenja preko raspoloživih uređaja kao što su Micro:bit, arduino...).

**DIDAKTIČKO-METODIČKA UPUTSTVA preuzeta iz Plana i programa predmeta Informatika sa tehnikom.*



ČASOVI PROGRAMIRANJA – MATERIJALI ZA UČENJE

Prilikom pripreme materijala za realizaciju nastave, nastavnik treba da izbjegava klasična predavanja. Časove treba organizovati kao neku vrstu radionice, a scenario časa kreirati tako da svi učenici neprekidno učestvuju u razvoju programa za Micro:bit i demonstraciji algoritama sortiranja.

Predlažemo nekoliko scenarija časa koja obuhvataju ishode učenja, kao i neke naprednije ishode koji nijesu obuhvaćeni nastavnim programom i mogu biti upotrijebljeni za dodatnu nastavu ili Micro:bit klub.

Dodatna oprema može se koristiti u predloženim projektima, bilo kao dio redovne nastave ili kao podsticaj za one učenike koji žele više.

Svi časovi su organizovani tako da nekoliko prvih aktivnosti predstavljaju obnavljanje i utvrđivanje usvojenih vještina i znanja. Predloženo je više varijanti časa za sortiranje elemenata niza. Nastavnik po sopstvenoj procjeni može izabrati koju varijantu želi. Za utvrđivanje mogu se iskoristiti aktivnosti 1–5 sa časa 2 i časovi 3 i 6.

PLAN ČASA BROJ 1

OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da objasni potrebu za upotrebom nizova u programima i kreira nizove u programu.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, uređaj Micro:bit.

KRATAK OPIS ČASA

Učenici se upoznaju sa nizovima. Niz je promjenljiva koja čuva više vrijednosti istog tipa. Na primjer, niz ocjene, koji sadrži 5 brojeva koji predstavljaju ocjene iz matematike, fizike, hemije, biologije i informatike: 5, 4, 5, 4, 3. Brojevi 5, 4, 5, 4 i 3 su elementi niza. Kao drugi primjer, posmatrajmo niz „predmeti”, koji sadrži 5 stringova koji predstavljaju nazive predmeta: „Matematika”, „Fizika”, „Hemija”, „Biologija” i „Informatika”. Svakom elementu pridružen je jedan broj koji se naziva indeks elementa i koji označava poziciju elementa u nizu.

Indeksi u MakeCode-u, JavaScript-u i Python-u počinju od broja 0, kako je prikazano na slici:

Ocjene

Indeks	0	1	2	3	4
Element	5	4	5	4	3

Predmeti

Indeks	0	1	2	3	4
Element	„Matematika”	„Fizika”	„Hemija”	„Biologija”	„Informatika”

Na primjer, `ocjene[4]` predstavljaju element niza 3, dok `predmeti[2]` predstavljaju string (tekst) „Hemija”.

Učenici će kreirati nizove brojeva i stringova, pristupati elementima niza i modifikovati niz.

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Demonstracija primjera nizova – Učenici slušaju nastavnikovo izlaganje o pojmu niza i primjerima nizova. Zapisuju primjere nizova ocjena i predmeta i tabele koje opisuju vezu između indeksa i elementa niza. Odgovaraju na pitanja nastavnika:

- Koliko elemenata ima dati niz?
- Koji element ima indeks 3?
- Da li niz može imati iste elemente?
- Koji je indeks elementa „Hemija”?

Diskutuju o primjerima nizova sa kojima se svakodnevno susreću (npr. niz ocjena iz matematike na testu, niz vremena u trci Formule 1...).

Zapisuju druge nazive za niz: JavaScript koristi engleski pojam *array*, dok MakeCode i Python koriste termin *list*.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik objašnjava učenicima zahtjeve zadatka koje će raditi u nastavku. Potrebno je napraviti kôd koji će pri pokretanju kreirati niz *ocjene* sa sljedećim ocjenama: [5, 4, 4, 3, 4, 5]. Pritiskom na taster A, prikazati ocjenu čiji je indeks 4. Pritiskom na dugme B, promijenimo ocjenu sa indeksom 4 („pokvariće” ocjenu sa 3 na 2). Pritiskom na dugme A+B, dodaćemo novu ocjenu 4, kao posljednji element niza, prikazati taj element niza i zatim prikazati dužinu niza, tj. broj elemenata u nizu.

Aktivnost 3

Otvaranje sekcije za nizove (Advanced/Arrays) – Učenici, prateći instrukcije nastavnika, u okviru programa MakeCode otvaraju sekciju za nizove. Klikom na *Advanced*, otvara se lista dodatnih potkategorija i bira se *Arrays*. Učenici biraju blok *set list to array...*



Mijenjaju ime niza iz zadatog *list* u *ocjene*.

Pregledaju sve podsekcije u sekciji *Arrays*: *Create*, *Read*, *Modify* i *Operations*.



Aktivnost 4

Dodavanje novih elemenata niza – Učenici u okviru bloka *set list to array...*, klikom na +, dodaju nove elemente niza.



Niz *ocjene* sadrži 6 elemenata, koje učenici zapisuju na sljedeći način: [5, 4, 4, 3, 4, 5]. Nastavnik podsjeća učenike da će indeksi niza *ocjene* imati vrijednosti 0, 1, 2, 3, 4 i 5. Na primjer, pišemo *ocjene[0] = 5* ili *ocjene[4] = 4*.

U slučaju pogrešnog unosa, učenici mogu koristiti dugme „-“ za brisanje elemenata niza.



Aktivnost 5

Pristup i izmjena elemenata niza – Učenici dodaju blok *on button A pressed* i u njega dodaju blok *show number*. Slušaaju instrukcije nastavnika o bloku *list get value at*, a zatim dodaju taj blok u *show number*. Mijenjaju ime niza u tom bloku sa *list* na *ocjene*. Testiraju izvršavanje programa na simulatoru, mijenjajući indeks elemenata niza. Odgovaraju na pitanje nastavnika: „Šta se dobija ako postavite nepostojeći indeks (npr. -1 ili 10)?“



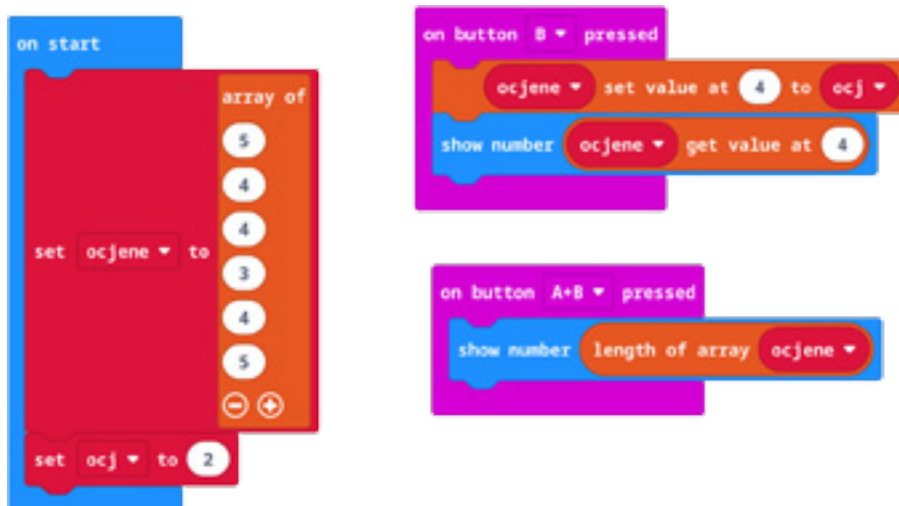
Aktivnost 6

Učenici dodaju blok *on button B pressed* i u njega dodaju blok *show number*. Slušaaju instrukcije nastavnika o bloku *list set value at ... to ...*, a zatim dodaju taj blok prije bloka *show number*. Mijenjaju ime niza u tom bloku sa *list* na *ocjene*. Kreiraju novu promjenljivu „ocj“ i u bloku *on start* postavljaju njenu vrijednost na 2. Mijenjaju blok *list set value at ... to ...* u *ocjene set value at 4 to ocj*. U blok *show number* dodaju blok *ocjene get value at 4*. Testiraju izvršavanje programa na simulatoru, mijenjajući vrijednost promjenljive „ocj“.

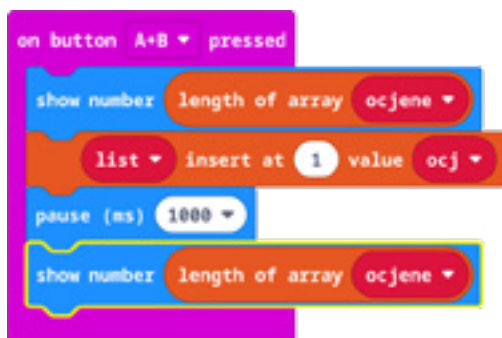


Aktivnost 7

Dodavanje elementa u niz – Učenici dodaju blok *on button A+B pressed* i u njega dodaju blok *show number*. Slušaaju instrukcije nastavnika o bloku *length of array list*, koji daje dužinu liste, tj. broj elemenata niza, a zatim dodaju taj blok u blok *show number*. Mijenjaju ime niza u tom bloku sa *list* na *ocjene*. Testiraju izvršavanje programa na simulatoru.



Slušaju instrukcije nastavnika o blokovima *list insert ... at beginning* i *list insert at ... value ...*. Prvi blok dodaje element na početak liste, a drugi dodaje element na datu poziciju poslije „at“. Učenici dodaju blok *ocjene insert at 1 value ocj*, pauzu od 1 sekunde i zatim blok *show number*, koji ponovo prikazuje dužinu niza *ocjene*. Testiraju izvršavanje programa na simulatoru.



Aktivnost 8

Niz stringova – Učenici kreiraju novi projekat u MakeCode-u, sa nazivom „Registarske tablice“. Kreiraju novi niz primjenom bloka *set text list to array of ...*



Mijenjaju ime niza u bloku u *tablice* i zadaju mu elemente: PG, NK, BR, BP, KL, na isti način kao i u slučaju niza brojeva. Slušaju saopštenje nastavnika o tome da se stringovi zapisuju sa navodnicima.

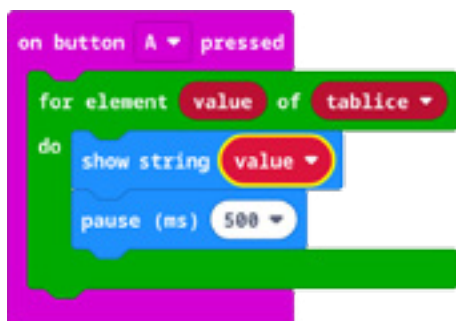


Aktivnost 9

Prikaz niza – Učenici slušaju saopštenje nastavnika o cilju zadatka: klikom na dugme A, potrebno je prikazati redom sve registarske tablice, sa pauzom od 500 milisekundi između dvije tablice. Učenici dodaju blok *on button A pressed*. Svim elementima niza može se pristupiti ili ciklusom koji prolazi (iterira) kroz sve elemente niza. Ako koristimo indekse, tada koristimo blok *list get value at ...*, navodeći indeks elementa. Postoji i drugi blok koji je namijenjen za iteraciju kroz niz bez upotrebe indeksa: blok *for element value of list*. U blok *on button A pressed* učenici dodaju blok *for element value of list* i mijenjaju ime niza iz *list* u *tablice*. Unutar tog bloka dodaju dva bloka: *show string*, u kome postavljaju promjenljivu *value* dovlačenjem iz bloka *for*, i blok *pause 1000*.

Aktivnost 10

Za one koji žele više – Aktivnost 9 realizovati kreiranjem funkcije *stampaTablica*, koja ima jedan argument *arr* tipa *Array*. Klikom na dugme A, pozvati ovu funkciju sa argumentom „tablice“.



Aktivnost 11

Testiranje – Učenici testiraju program na simulatoru i uklanjaju moguće greške. Prenose testirane programe na Micro:bit uređaj.

Aktivnost 12

Uraditi kviz na stranici <https://petlja.org/biblioteka/r/lekcije/microbitbc/%D1%87%D0%B0%D1%81-14---%D0%BD%D0%B8%D0%B7%D0%BE%D0%B2%D0%B8>

Program možete pogledati kada skenirate QR kôd ili na linku: https://makecode.microbit.org/_8vTLcJ4xhErD



PLAN ČASA BROJ 2	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da koristi osnovne operacije sa nizovima i kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, uređaj Micro:bit.
KRATAK OPIS ČASA	
Učenici se upoznaju sa nizovima brojeva, stringova i objekata. Kreiraju programe koji štampaju elemente niza.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Demonstracija primjera nizova – Učenici diskutuju o primjerima nizova *stringova* sa kojima se svakodnevno susreću (npr. niz naziva nastavnih predmeta, niz prezimena učenika u odjeljenju...). Za dati niz ocjena učenici nabrajaju mehanizme iteracije ocjena.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik objašnjava učenicima zahtjeve zadatka koje će raditi u nastavku. Potrebno je napraviti kôd koji će pri pokretanju kreirati niz *ocjene* sa sljedećim ocjenama: [5, 4, 4, 3, 4, 5]. Pritiskom na taster A, prikazati sve ocjene primjenom bloka *for element value of list*, sa pauzom od po 500 milisekundi između ocjena. Pritiskom na dugme B, uradiće istu stvar, ali primjenom bloka *for index from 0 to...* Pritiskom na dugme A+B, uradiće i treći put istu stvar, ali sada primjenom bloka *repeat ... times*.

Aktivnost 3

Otvaranje sekcije za nizove (Advanced/Arrays) – Učenici, prateći instrukcije nastavnika, u okviru programa MakeCode otvaraju sekciju za nizove. Ako su učenici uspješno realizovali aktivnost sa prvog časa, mogu kopirati kôd. Ako nijesu, ponavljaju postupak sa prvog časa. Klikom na *Advanced*, otvara se lista dodatnih potkategorija i bira se *Arrays*. Biraju blok *set list to array...*



Mijenjaju ime niza iz zadatog *list* u *ocjene*.



Aktivnost 4

Dodavanje novih elemenata niza – Učenici u okviru bloka *set list to array...*, klikom na +, dodaju nove elemente niza.



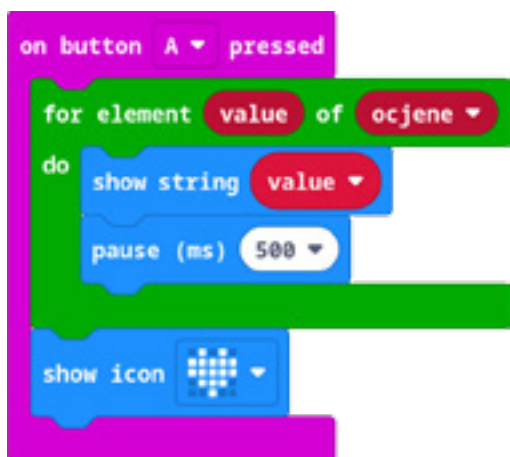
Niz *ocjene* sadrži 6 elemenata, koje učenici zapisuju na sljedeći način: [5, 4, 4, 3, 4, 5]. Nastavnik podsjeća učenike da će indeksi niza *ocjene* imati vrijednosti 0, 1, 2, 3, 4 i 5.

U slučaju pogrešnog unosa, učenici mogu koristiti dugme „-“ za brisanje elemenata niza.



Aktivnost 5

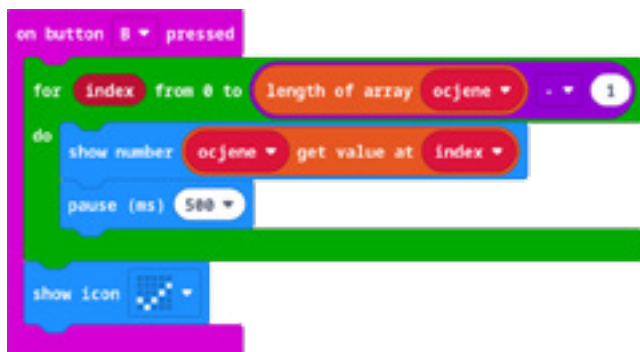
Prva verzija prikaza niza – Učenici dodaju blok *on button A pressed*, u njega dodaju blok *for element value of list* i mijenjaju ime niza iz *list* u *ocjene*. Unutar tog bloka dodaju dva bloka: *show string*, u kome postavljaju promjenljivu *value* dovlačenjem iz bloka *for*, i blok *pause 500*. Iza ciklusa *for element* dodaju blok *show icon* sa ikonom *heart*, što će označavati kraj prikaza elemenata niza.



Aktivnost 6

Druga verzija prikaza niza – Učenici dodaju blok *on button B pressed*, u njega dodaju blok *for index from 0 to 4*. U tom bloku zamijenimo broj 4 posljednjim indeksom niza *ocjene*, tj. sa *(length of array ocjene) - 1*. Unutar bloka *for* prvo dodaju blok za prikaz elementa niza „*ocjene*“, čiji je indeks zadana promjenljiva „*index*“. Kombinuju blok *show number* i *list get value at ...*, pri čemu u drugom bloku umjesto „*list*“ postavljaju „*ocjene*“, a u dijelu „*at*“ prevlače promjenljivu „*index*“. Zatim dodaju i blok za pauzu od 500 milisekundi.

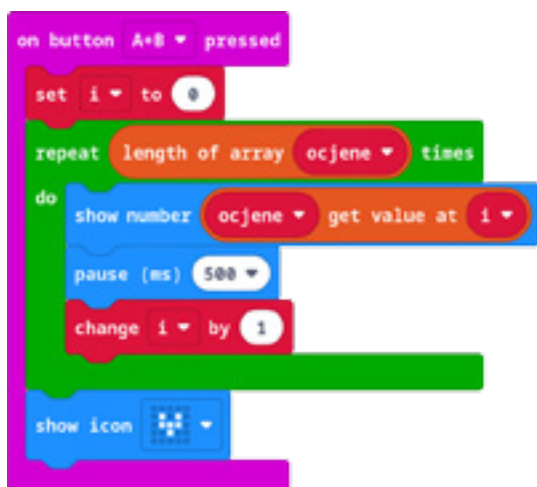
Testiraju izvršavanje programa na simulatoru.



Aktivnost 7

Treća verzija prikaza niza – Učenici prvo kreiraju novu promjenljivu „*i*“, koja će označavati indekse niza. Dodaju blok *on button A+B pressed* i u njega dodaju blok *set i to 0* i ciklus *repeat ... times*. Mijenjaju blok *repeat* tako da se izvrši *length of array ocjene* onoliko puta koliko imamo elemenata niza. Unutar bloka „*repeat*“ ponavljaju blokove iz prethodne aktivnosti: kombinuju blok *show number* i *list get value at ...*, pri čemu u drugom bloku umjesto „*list*“ postavljaju „*ocjene*“, a u dijelu „*at*“ prevlače promjenljivu „*i*“. Na kraju ciklusa dodaju blok koji uvećava „*i*“ za 1, tj. blok *change i by 1*. Poslije ciklusa opet dodaju blok *show icon* sa proizvoljnom slikom koja nije „*heart*“.

Testiraju izvršavanje programa na simulatoru.



Aktivnost 8

Za one koji žele više: Napisati tri različite funkcije koje implementiraju operacije iz aktivnosti 5, 6 i 7, a zatim dodati kôd koji pritiskom na odgovarajuće dugme poziva te funkcije. Učenici sami imenuju funkcije i zadaju im parametre.

Aktivnost 9

Učenici prenose testirani program na Micro:bit uređaj.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_Hk1LmKYVhiai



PLAN ČASA BROJ 3	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, uređaj Micro:bit.
KRATAK OPIS ČASA	
Učenici kreiraju programe koji nalaze prosječnu ocjenu i ukupan broj ocjena jednakih 5.	
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA	

Aktivnost 1

Demonstracija primjera nizova – Učenici zapisuju dati niz ocjena i ručno utvrđuju koja je prosječna ocjena i koliko ima ocjena „odličan”.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik objašnjava učenicima zahtjeve zadatka koje će raditi u nastavku. Potrebno je napraviti kôd koji će pri pokretanju kreirati niz *ocene* sa sljedećim ocjenama: [5, 4, 4, 3, 4, 5]. Pritiskom na taster A, prikazati prosječnu ocjenu. Pritiskom na dugme B, štampaće koliko ima ocjena „odličan”. Pritiskom na dugme A+B, svira melodiju zadatu nizom nota.

Aktivnost 3

Otvaranje sekcije za nizove (Advanced/Arrays) – Učenici, prateći instrukcije nastavnika, u okviru programa MakeCode otvaraju sekciju za nizove. Ako su učenici uspješno realizovali aktivnost sa prvog časa, mogu kopirati kôd. Ako nijesu, ponavljaju postupak sa prvog časa. Klikom na *Advanced*, otvara se lista dodatnih potkategorija i bira se *Arrays*. Biraju blok *set list to array...*



Mijenjaju ime niza iz zadanog *list* u *ocjene*.



Aktivnost 4

Dodavanje novih elemenata niza – Učenici u okviru bloka *set list to array...*, klikom na +, dodaju nove elemente niza.



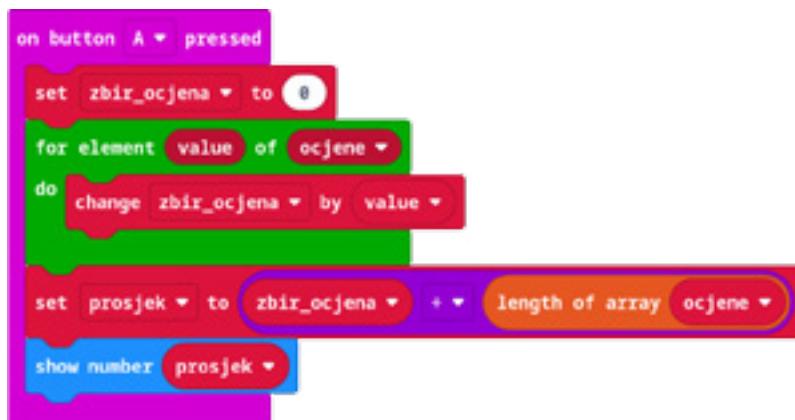
Niz *ocjene* sadrži 6 elemenata, koje učenici zapisuju na sljedeći način: [5, 4, 4, 3, 4, 5]. Nastavnik podsjeća učenike da će indeksi niza *ocjene* imati vrijednosti 0, 1, 2, 3, 4 i 5.

U slučaju pogrešnog unosa, učenici mogu koristiti dugme „-“ za brisanje elemenata niza. Učenici ogovaraju na pitanje: „Kako se izračunava prosjek?“ i zatim dodaju dvije promjenljive: „zbir_ocjena“ i „prosjek“.



Aktivnost 5

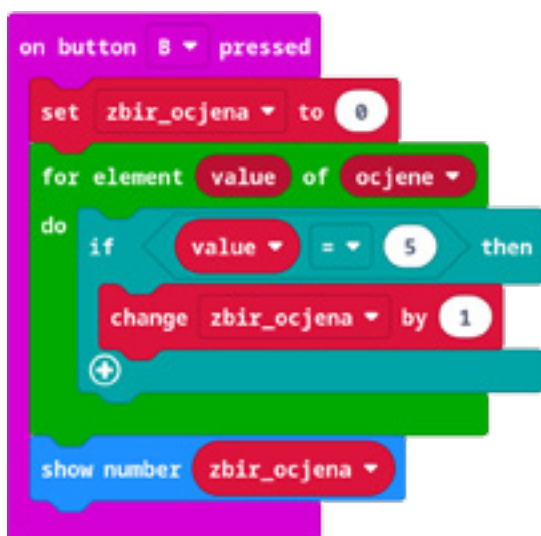
Prosjek ocjena – Učenici dodaju blok *on button A pressed* i u njemu postavljaju promenljivu „zbir_ocjena na 0“. Zatim dodaju blok *for element value of list*, mijenjaju ime niza iz *list* u *ocjene* i unutar tog bloka dodaju blok *change zbir_ocjena by value*. Po završetku ciklusa, dodaju blok koji promjenljivoj „prosjek“ dodjeljuje vrijednost količnika promjenljive „zbir_ocjena“ i broja elemenata niza (tj. bloka *length of array ocjene*). Na kraju, učenici dodaju blok *show number prosjek*, koji prikazuje dobijeni rezultat.



Testiraju izvršavanje programa na simulatoru.

Aktivnost 6

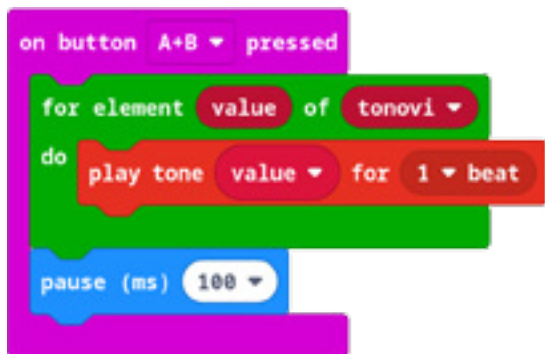
Iteracija kroz niz – Učenici dodaju blok *on button B pressed* i u njemu postavljaju promjenljivu „zbir_ocjena na 0“. Slušaju objašnjenje nastavnika da se ista promjenljiva može više puta upotrijebiti u programu. U ovom slučaju, promjenljiva „zbir_ocjena“ imaće ulogu brojača, tj. u njoj će čuvati koliko ima petica u nizu *ocjene*. Zatim dodaju blok *for element value of list* i mijenjaju ime niza iz *list* u *ocjene*. U bloku *for* pomoću bloka *if value = 5* ispituju da li je zadovoljen uslov. Ako jeste, unutar tog bloka uvećavaju brojač „zbir_ocjena“ za 1 pomoću bloka *change zbir_ocjena by 1*. Po završetku ciklusa, dodaju blok *show number zbir_ocjena*, koji prikazuje dobijeni rezultat.



Testiraju izvršavanje programa na simulatoru.

Aktivnost 7

Niz tonova – Učenici slušaju pojašnjenje nastavnika da se note mogu prikazati brojevima. Kreiraju novi niz *tonovi* i postavljaju mu sljedeće vrijednosti: [262, 392, 330, 392, 262]. U blok *on start* dodaju niz *tonovi*. Dodaju blok *on button A+B pressed*, u koji dodaju blok *for element value of list* i mijenjaju ime niza iz *list* u *tonovi*. Unutar bloka *for* dodaju blok *play tone value for 1 beat*, a poslije bloka *for* i *pause (ms) 100*.



Testiraju program na simulatoru.

Aktivnost 8

Učenici prenose testirani program na Micro:bit uređaj.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_ErLYT7TgKXXt



PLAN ČASA BROJ 4	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.
KRATAK OPIS ČASA	
Učenici se upoznaju sa načinima sortiranja niza. Naučiće tri jednostavna metoda sortiranja: BubbleSort, selection sort i insertion sort. Sortirati niz znači poređati njegove članove u nekom poretku: rastućem, opadajućem, strogo rastućem ili strogo opadajućem.	
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA	

Aktivnost 1

Demonstracija algoritma BubbleSort – Pet učenika dobija po jedan list papira. Na svakom listu papira je zapisan po jedan element niza [5, 1, 4, 2, 8]. Učenici prate instrukcije nastavnika. Sve dok niz nije sortiran, ponavljaju sljedeće radnje: svaka dva susjedna učenika koja nijesu u pravom poretku mijenjaju mjesta. Odgovaraju na pitanje: „Koliko je ukupno zamjena potrebno u najmanje povoljnom slučaju?”

Aktivnost 2

Učenici prate izlaganje nastavnika o Bubble sort-u. Zapisuju primjer niza sa 5 elemenata: [5, 1, 4, 2, 8]. Upoređuju susjedne elemente niza i ako nijesu u odgovarajućem poretku, zamjenjuju im mjesta. Ovaj postupak ponavljaju sve dok ima zamjene mjesta u nizu. Sortiranje se obično obavlja u samom nizu (engl. *in-place*), bez uvođenja pomoćnog niza. Učenici zapisuju korake algoritma:

Prvi prolaz:

[5, 1, 4, 2, 8] → [1, 5, 4, 2, 8] – zamjena prva dva elementa jer je $5 > 1$
 [1, 5, 4, 2, 8] → [1, 4, 5, 2, 8] – zamjena drugog i trećeg elementa jer je $5 > 4$
 [1, 4, 5, 2, 8] → [1, 4, 2, 5, 8] – zamjena jer je $5 > 2$
 [1, 4, 2, 5, 8] → [1, 4, 2, 5, 8] – nema zamjene jer je $8 > 5$

Drugi prolaz:

[1, 4, 2, 5, 8] → [1, 4, 2, 5, 8]
 [1, 4, 2, 5, 8] → [1, 2, 4, 5, 8] – zamjena jer je $4 > 2$
 [1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]
 [1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

Treći prolaz:

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]
 [1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]
 [1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]
 [1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

Nije bilo zamjene mjesta, pa algoritam završava rad.

Aktivnost 3

Učenici zapisuju algoritam, koji se može opisati sljedećim pseudokodom, gdje je A niz koji ima n elemenata sa indeksima 0, ..., n-1:

```
function Bubble_sort A
    n ← length(A)
    repeat
        swapped ← false /* da li se dogodila zamjena mjesta elemenata */
        for i ← 1 to n-1 do
            /* ako elementi nijesu u pravom poretku */
            if A[i-1] > A[i] then
                /* zamijeni mjesta A[i-1] i A[i] */
                t ← A[i-1]
                A[i-1] ← A[i]
                A[i] ← t
                swapped ← true
            end if
        end for
    until not swapped
```

Umjesto datog pseudokoda, nastavnik može, zajedno sa učenicima, nacrtati dijagram toka algoritma.

Aktivnost 4

Učenici slušaju uvodno izlaganje nastavnika o sortiranju umetanjem (Insertion sort): algoritam uklanja elemente jedan po jedan iz nesortiranog niza i umeće ih u sortirani dio niza na odgovarajuću poziciju. Sortiranje se obično obavlja u samom nizu (engl. *in-place*), bez uvođenja pomoćnog niza. Elementi se uklanjaju sa početka ili kraja niza, čime se stvara prostor za sortirani dio niza.

Aktivnost 5

Učenici zapisuju primjer sortiranja niza sa 8 elemenata primjenom algoritma *Insertion sort*. Elementi označeni žutom bojom čine sortirani dio niza, dok bijela boja označava elemente u nesortiranom dijelu niza. Crvena boja označava elemente koji su pomjereni u trenutnom koraku. Umjesto žute boje, učenici podvlače elemente, a umjesto crvene boje zaokružuju ih.

Niz	Element koji umećemo	Niz poslije umetanja
[12, 8, 23, 6, 34, 22, 45, 9]	8	[8, 12, 23, 6, 34, 22, 45, 9]
[8, 12, 23, 6, 34, 22, 45, 9]	23	[8, 12, 23, 6, 34, 22, 45, 9]
[8, 12, 23, 6, 34, 22, 45, 9]	6	[6, 8, 12, 23, 34, 22, 45, 9]
[6, 8, 12, 23, 34, 22, 45, 9]	34	[6, 8, 12, 23, 34, 22, 45, 9]
[6, 8, 12, 23, 34, 22, 45, 9]	22	[6, 8, 12, 22, 23, 34, 45, 9]
[6, 8, 12, 22, 23, 34, 45, 9]	45	[6, 8, 12, 22, 23, 34, 45, 9]
[6, 8, 12, 22, 23, 34, 45, 9]	9	[6, 8, 9, 12, 22, 23, 34, 45]

Aktivnost 6

Učenici zapisuju algoritam sortiranja umetanjem (*Insertion sort*). Ako je A niz od n elemenata koji sortiramo, čiji su indeksi 0, ..., n-1, tada se ovaj algoritam može opisati sljedećim pseudokodom:

```
function insertion_sort A
    i ← 1
    while i < n do
        v ← A[i]
        h = i
        /* odredimo poziciju gdje umećemo Član v */
        while (h > 0 and A[h-1] > v)
            A[h] ← A[h-1]
            h ← h - 1
        /* umećemo v na poziciju h */
        A[h] ← v;
        i ← i + 1
```

Aktivnost 7

Učenici slušaju izlaganje nastavnika o sortiranju selekcijom (engl. *selection sort*). Metod se zasniva na podjeli niza na sortirani i nesortirani dio, na pronalaženju najmanjeg elementa u nesortiranom dijelu niza i zamjeni mjesta elemenata niza. Zapisuju algoritam za ovaj metod sortiranja:

1. Odredi najmanji elemenat u nizu. Zamijeni mjesta prvom elementu niza, tj. onom sa indeksom 0, i najmanjem elementu niza.
2. Odredi drugi najmanji elemenat u nizu. Zamijeni mjesta drugom elementu niza i drugom najmanjem elementu niza.
3. Odredi treći najmanji elemenat u nizu. Zamijeni mjesta trećem elementu niza i trećem najmanjem elementu niza.
4. Ponavljaj postupak određivanja sljedećeg najmanjeg elementa i zamjene mjesta, sve dok niz ne bude uređen.

Aktivnost 8

Demonstracija algoritma sortiranja – Osam učenika dobija po jedan list papira. Na svakom listu papira je zapisan po jedan element niza [12, 8, 23, 6, 34, 22, 45, 9]. Učenici stoje u vrsti i prate instrukcije nastavnika. Deveti učenik, označimo ga sa AAA, stoji ispred prvog učenika, leđima prema učionici i pokazuje prstom na prvog učenika. Ostali učenici traže koji od učenika iz vrste, koji se nalaze desno od AAA, drži najmanji broj, tj. na kojoj se poziciji u vrsti nalazi najmanji broj, i izgovaraju glasno njegov indeks, koji označimo sa IND. Tada učenik na koga pokazuje AAA i učenik IND mijenjaju mjesta. Zatim se AAA pomjeri tako da pokazuje na sljedećeg učenika u vrsti.

Ostali učenici ponavljaju postupak: traže koji od učenika iz vrste, koji se nalaze desno od AAA, drži najmanji broj, tj. na kojoj se poziciji u vrsti nalazi najmanji broj, i izgovaraju glasno njegov indeks, koji označimo sa IND. Tada učenik na koga pokazuje AAA i učenik IND mijenjaju mjesta.

Ovaj postupak se ponavlja sve dok učenik AAA ne dođe do posljednjeg učenika u vrsti. Učenici zaključuju da učenici u vrsti sada drže papire koji prikazuju rastući niz brojeva.

Aktivnost 9

Učenici odgovaraju na pitanja nastavnika: „Koji je najmanji element u nesortiranom dijelu niza?“, „Koji je indeks tog najmanjeg elementa?“ i „Kako izgleda niz poslije zamjene mjesta?“ i popunjavaju tabelu koja prikazuje korake algoritma sortiranja selekcijom, na konkretnom nizu [12, 8, 23, 6, 34, 22, 45, 9]:

Niz	Najmanji element	Niz poslije zamjene
[12, 8, 23, 6, 34, 22, 45, 9]	6	[6, 8, 23, 12, 34, 22, 45, 9]
[6, 8, 23, 12, 34, 22, 45, 9]	8	[6, 8, 23, 12, 34, 22, 45, 9]
[6, 8, 23, 12, 34, 22, 45, 9]	9	[6, 8, 9, 12, 34, 22, 45, 23]
[6, 8, 9, 12, 34, 22, 45, 23]	12	[6, 8, 9, 12, 34, 22, 45, 23]
[6, 8, 9, 12, 34, 22, 45, 23]	22	[6, 8, 9, 12, 22, 34, 45, 23]
[6, 8, 9, 12, 22, 34, 45, 23]	23	[6, 8, 9, 12, 22, 23, 45, 34]
[6, 8, 9, 12, 22, 23, 45, 34]	34	[6, 8, 9, 12, 22, 23, 34, 45]
[6, 8, 9, 12, 22, 23, 34, 45]	45	[6, 8, 9, 12, 22, 23, 34, 45]
[6, 8, 9, 12, 22, 23, 34, 45]		

PLAN ČASA BROJ 5	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da demonstrira program koji povezuje uređaje.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit uređaj.
KRATAK OPIS ČASA	
Učenici se upoznaju sa mogućnostima radio-veze u Micro:bit-u. Definišu pojmove <i>predajnik</i> (engl. <i>Sender</i>) i <i>prijemnik</i> (engl. <i>Receiver</i>) i implementiraju programe koji šalju brojeve, stringove i parove „naziv – vrijednost”. Kreiraju program kojim se pomoću Micro:bit-a upozorava na držanje rastojanja između dva učenika.	
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA	

Aktivnost 1

Demonstracija radio-veze – Učenici odgovaraju na pitanje: „Da li ste imali ili imate voki-toki uređaj?” Opisuju princip rada voki-toki uređaja: jedan govori, a ostali slušaju. Zapisuju pojmove *prijemnik* i *predajnik* i uočavaju da se isti uređaj može koristiti i kao prijemnik i kao predajnik. Nabrajaju klasične radio-stanice koje znaju (ne internet radio-stanice). Odgovaraju na pitanje nastavnika: „Ko emituje vijesti i muziku, a ko je prima?” Diskutuju o mogućim odgovorima. Opisuju prijemnik i predajnik. Izvršavaju uputstva nastavnika kako da formiraju parove koji će zajednički pisati programe i testirati ih.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa sekcijom Radio i opisuje zadatke koje će raditi u nastavku, a koje učenici zapisuju. Potrebno je napraviti kôd koji će pritiskom na taster A poslati string svim ostalim Micro:bit uređajima u grupi.

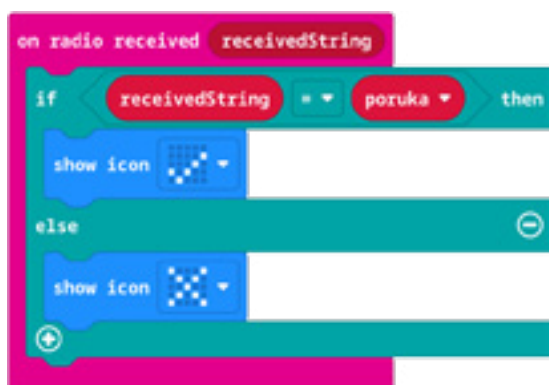
Aktivnost 3

Program za slanje poruke (send) – Učenici kreiraju promjenljivu „poruka” u bloku *on start* i postavljaju njenu vrijednost na „Hello”. Prate instrukcije nastavnika i u okviru programa MakeCode otvaraju sekciju Radio, iz koje u blok *on start* dodaju blok *radio set group 1*. Svaki par učenika mora imati različitu grupu od 1 do 255, dok uređaji u grupi imaju isti broj. Dodaju blok *on button A pressed* i u njemu šalju vrijednost promjenljive „poruka” svim uređajima u grupi primjenom bloka *radio send string poruka*. Odgovaraju na pitanje nastavnika: „Koje sve tipove podataka je moguće poslati ugrađenom radio-vezom u Micro:bit-u?” Učenici testiraju program.



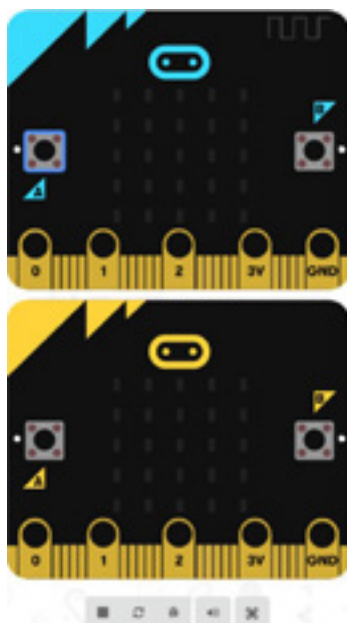
Aktivnost 4

Program za primanje poruke (receive) – Učenici dodaju blok *on radio received* *receivedString* i u njemu pomoću naredbe *if* provjeravaju da li je primljeni string upravo ono što je emitovano. Ako jeste, dodaju blok *show icon ...* sa jednom ikonom, a ako nije, drugi blok *show icon ...* sa nekom drugom ikonom.

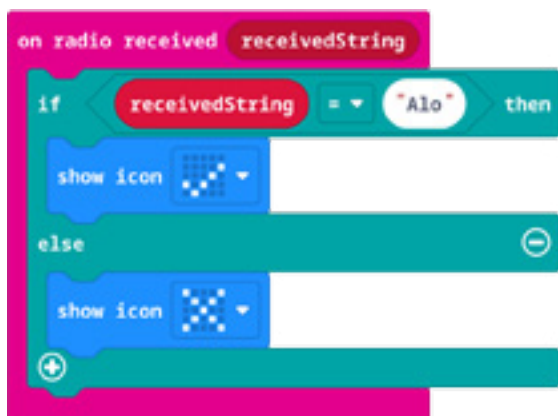


Testiraju program pojedinačno na simulatoru. Simulator dodaje sliku još jednog Micro:bit uređaja, klikom na dugme A jednog od uređaja, simulator prikazuje odgovarajuću ikonu na drugom uređaju.

Učenici mijenjaju program tako da umjesto provjere *receivedString = poruka* u bloku *if* unose uslov *receivedString = Alo* ili neki drugi string koji nije *Hello*. Pokretanjem programa i pritiskom na taster A, na prijemniku se prikazuje druga ikona.



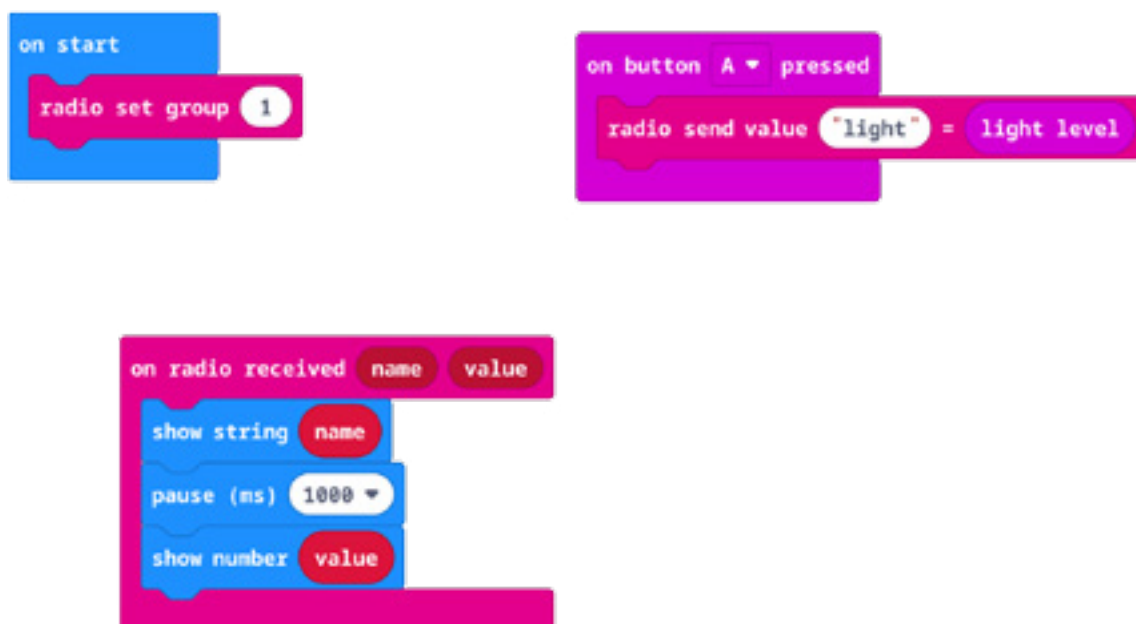
Ako nema grešaka u programu, oba učenika iz para prenose program na svoje uređaje i testiraju ponašanje programa.



Aktivnost 5

Parovi naziv – vrijednost – Učenici slušaju opis zadatka: napisati program koji pritiskom na dugme A šalje par „naziv – vrijednost“ (engl. *name – value*) svim uređajima u grupi. Npr. jedan par učenika šalje („light“ : light level), drugi par učenika šalje („ubrzanje“ : acceleration (mg) x), treći par učenika šalje („ubrzanje“ : acceleration (mg) y) itd.

Učenici kreiraju novi projekat i na „on start“ dodaju blok *radio set group 1*. Dodaju blok *on button A pressed* i u njega blok *radio send value 'light' light level* (ili blok *radio send value 'ubrzanje' acceleration (mg) x*, tj. odgovarajući blok za dati par učenika). Takođe, dodaju blok *on radio received value* i u njemu redom blokove *show string value*, *pause (ms) 1000* i *show number value*.



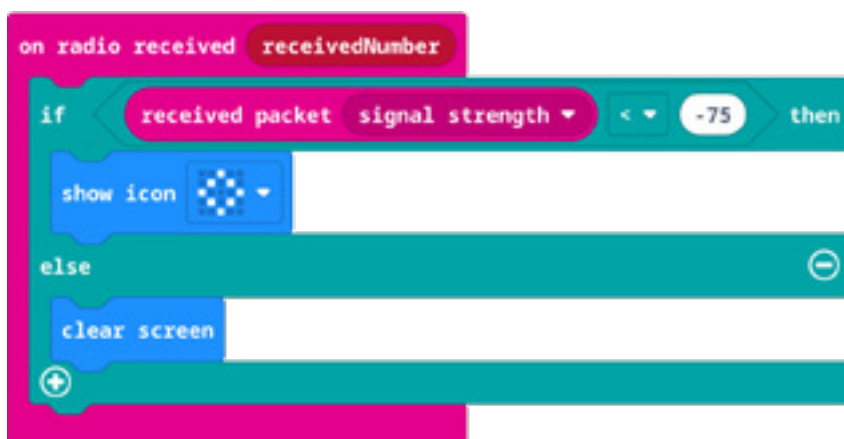
Aktivnost 6

Mjerenje udaljenosti radio-vezom – Učenici kreiraju novi projekat i slušaju opis zadatka: napisati program koji prikazuje ikonu kada se udaljimo od drugog učenika iz para. Ideja zadatka je da održavamo distancu između učenika pomoću Micro:bit uređaja.

U blok *on start* dodajemo *radio set group 1*, a u blok *forever* dodajemo blok *radio send number 0* (ili neki drugi broj).



Jedan od načina mjerenja udaljenosti između uređaja u grupi jeste i mjerenje jačine signala: što je signal jači, uređaji su bliži jedan drugom. Dodaju blok *on radio received name value* i u njemu naredbu *if* pomoću koje ispituju jačinu signala (engl. *received packet signal strength*). Jačina signala može imati vrijednosti između -128 i -28 (ili -42, u zavisnosti od verzije Micro:bit uređaja), gdje -128 znači slab signal, a -28 označava jak signal. Učenici eksperimentišu koju će jačinu signala postaviti kao graničnu (na slici je ta vrijednost -75). Ako je uslov tačan, prikazati ikonu po izboru pomoću bloka *show icon*, a ako nije, brišemo ekran uređaja blokom *clear screen*.



Aktivnost 7

Poslije provjere na simulatoru, učenici prenose program sa računara na Micro:bit uređaj. Testiraju program tako što se jedan učenik iz para udaljava od drugog učenika. Kada dostignu određenu udaljenost, oba uređaja počinju da prikazuju izabranu ikonu.

Programe možete pogledati kada skenirate QR kôdove ili na linkovima:

https://makecode.microbit.org/_LmxcV2eTucmq

https://makecode.microbit.org/_4JP57ecbA2ix

https://makecode.microbit.org/_fWjPd5HU4DPH



PLAN ČASA BROJ 6	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste radio i povezane računare za slanje i primanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit uređaj.
KRATAK OPIS ČASA	
Učenici pišu program koji kombinuje nizove i radio. Program šalje jednu sliku iz niza radio-vezom, pri čemu se tasterima A i B krećemo kroz niz slika – od prve slike ka posljednjoj ili od posljednje ka prvoj.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

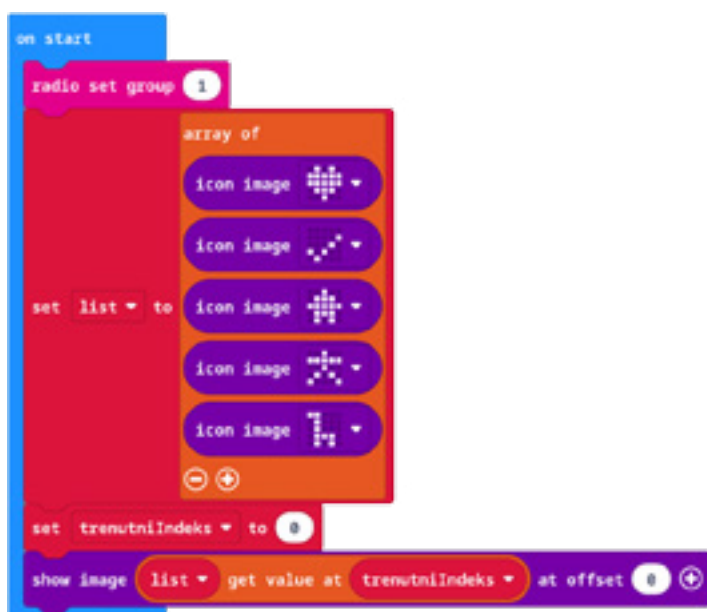
Aktivnost 1

Pokretanje programa – Učenici izvršavaju uputstva nastavnika kako da formiraju parove koji će zajednički pisati programe i testirati ih. Na svojim računarima pokreću program MakeCode i prate objašnjenja nastavnika o zadatku koji će raditi u nastavku, a koja učenici zapisuju. Potrebno je napraviti program koji će:

- pritiskom na taster A, prikazati sljedeću sliku iz niza;
- pritiskom na taster B, prikazati prethodnu sliku iz niza;
- pritiskom na taster A+B, poslati izabranu sliku ostalim Micro:bit uređajima u grupi.

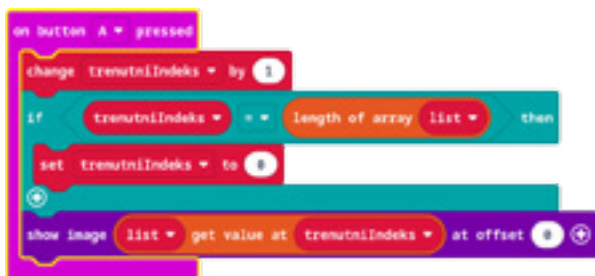
Aktivnost 2

Učenici kreiraju niz *list* i dodaju blok *set list to array ...* u blok *on start*. Dodaju i blok *radio set group 1*, gdje svaki par učenika ima svoju grupu. Primjenom bloka *icon image ...* u niz dodaju 5 različitih slika/ikona. Zatim kreiraju promjenljivu „trenutniIndeks” i u bloku *on start* postavljaju njenu vrijednost na 0. Kao posljednji blok u *on start* dodaju *show image* za prikaz elementa niza čiji je indeks *trenutniIndeks*. Dodaju i blokove za *on button A pressed*, *on button B pressed* i *on button A+B pressed*.



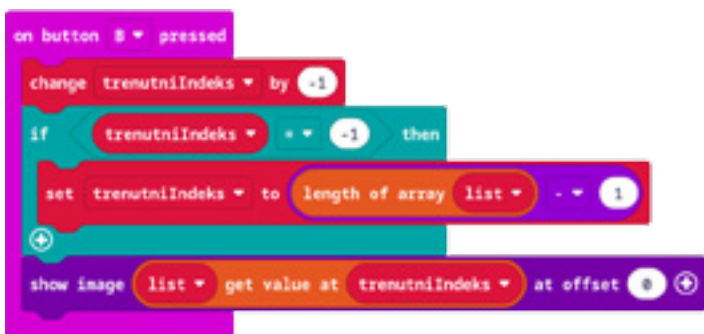
Aktivnost 3

Učenici pišu kôd za pomjeranje slike sa trenutne na sljedeću u nizu slika. U blok „on button A pressed”, prvo uvećavaju promjenljivu „trenutniIndeks” za 1. Pomoću *if* naredbe ispituju da li je ova promjenljiva „izašla” iz mogućih vrijednosti za indekse niza, tj. da li je jednaka dužini niza. Ako jeste, potrebno je postaviti „trenutniIndeks” na 0. Na kraju, prikazujemo sliku/ikonu čiji je indeks vrijednost promjenljive „trenutniIndeks”.



Aktivnost 4

Učenici pišu kôd za pomjeranje slike sa trenutne na prethodnu u nizu slika. U blok *on button B pressed*, prvo umanjuju promjenljivu „trenutniIndeks” za 1. Pomoću *if* naredbe ispituju da li je ova promjenljiva „izašla” iz mogućih vrijednosti za indekse niza, tj. da li je jednaka -1. Ako jeste, potrebno je postaviti *trenutniIndeks* na najveći mogući indeks, tj. (length of list) - 1. Na kraju, prikazujemo sliku/ikonu čiji je indeks vrijednost promjenljive „trenutniIndeks”.



Aktivnost 5

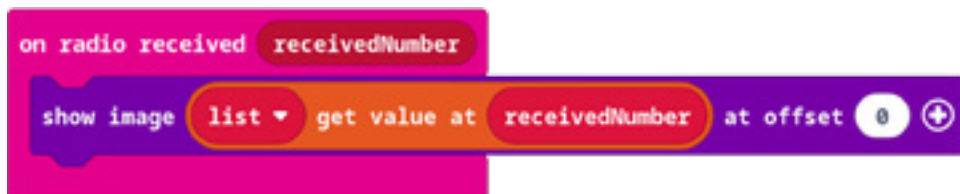
Za one koji žele više – Učenici kreiraju funkciju koja vraća sljedeći indeks; primjenom operacije *remainder of*, kreiraju kôd za prelazak na sljedeću sliku.

Aktivnost 6

U blok *on button A+B pressed* učenici dodaju blok *radio send number trenutniIndeks*. Odgovaraju na pitanje nastavnika: „Da li šaljemo sliku ili samo indeks slike, tj. broj?”



Dodaju blok *on radio received receivedString* i u njemu prikazuju sliku/ikonu čiji je indeks primljeni broj.



Aktivnost 7

Testiraju program pojedinačno na simulatoru. Ako nema grešaka u programu, oba učenika iz para prenose program na svoje uređaje i testiraju ponašanje programa. Moguće je kreirati i grupe sa više od 2 učenika.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_Tzf2YxiubPuV



PLAN ČASA BROJ 7	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da implementira programe koji upisuju podatke u fajl.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit uređaj.
KRATAK OPIS ČASA	
Učenici pišu programe koji serijskom vezom prenose podatke sa Micro:bit uređaja u fajlove koji se čuvaju na disku računara.	
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA	

Aktivnost 1

Učenici diskutuju o tipovima podataka koje su do sada koristili u Micro:bit-u. Zapisuju primjere različitih tipova podataka:

- tekst (string): npr. „Informatika”, „Hello Logo!”, „a”;
- brojevi: npr. 354 ili 0 ili 23.45;
- nizovi brojeva: npr. 11, 3, 45, 76;
- parovi (ime, vrijednost) (engl. *name – value pairs*): accelY:956.

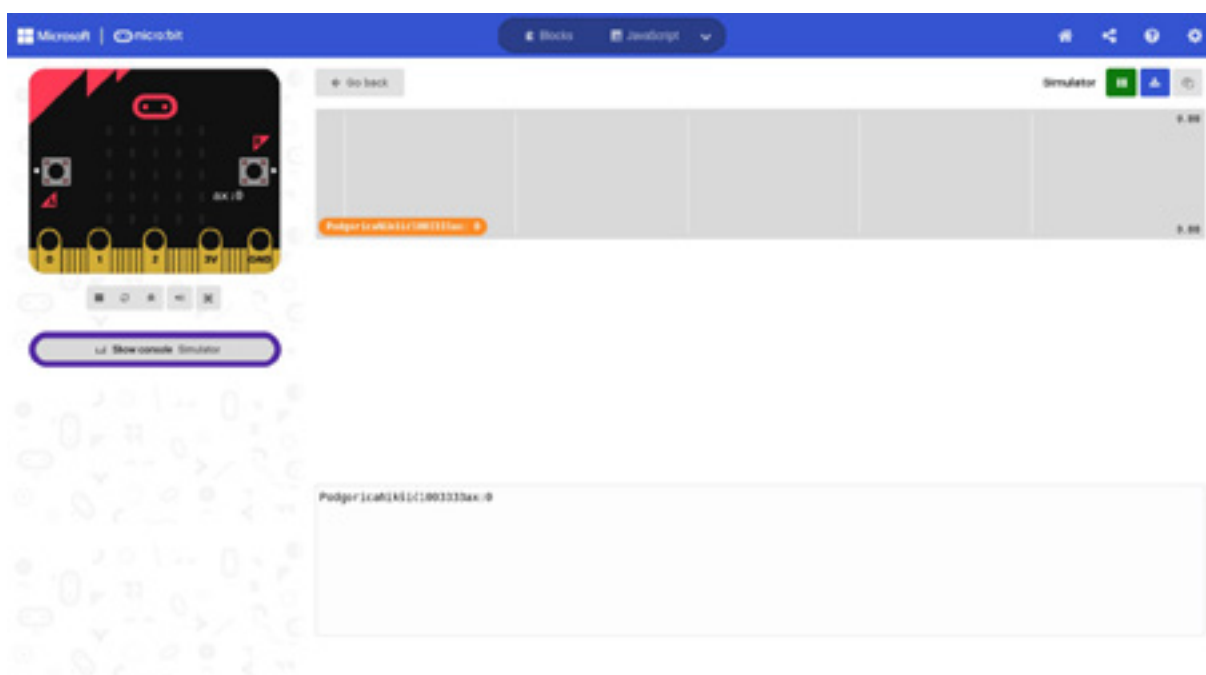
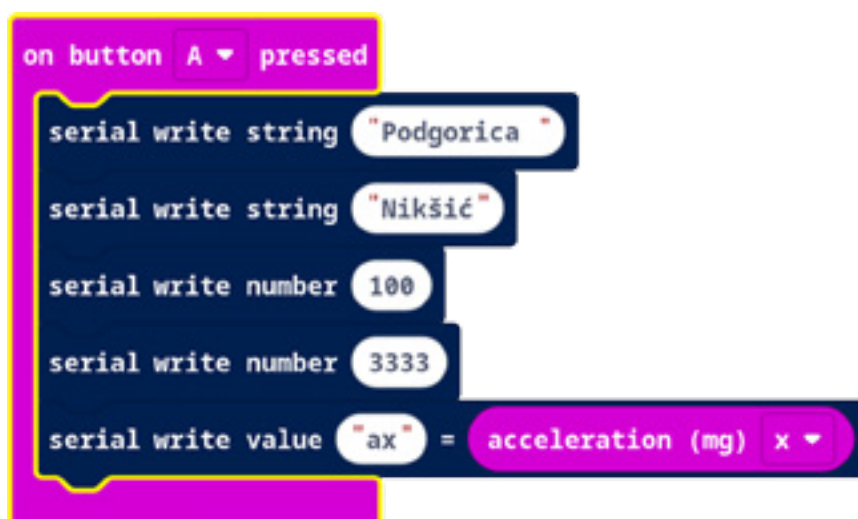
Slušaju izlaganje nastavnika o serijskom prenosu podataka. Svi podaci koji koriste takozvane serial funkcije u MakeCode-u se čuvaju, bez obzira na to da li koristimo simulator ili Micro:bit uređaj priključen na računar. Poseban dio Micro:bit-a, tzv. DataViewer, prepoznaje format podataka i odlučuje na koji će ih način prikazati.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode, kreiraju novi projekat i prate objašnjenja nastavnika. Dodaju blok *on button A pressed*, a zatim iz sekcije *Serial* dodaju 2 bloka *serial write string* sa stringovima „Podgorica” i „Nikšić”, zatim dva bloka *serial write number* sa brojevima 100 i 3333 i na kraju jedan blok *serial write value*, gdje upisuju par *ax* i blok *acceleration (mg) x*.

Aktivnost 3

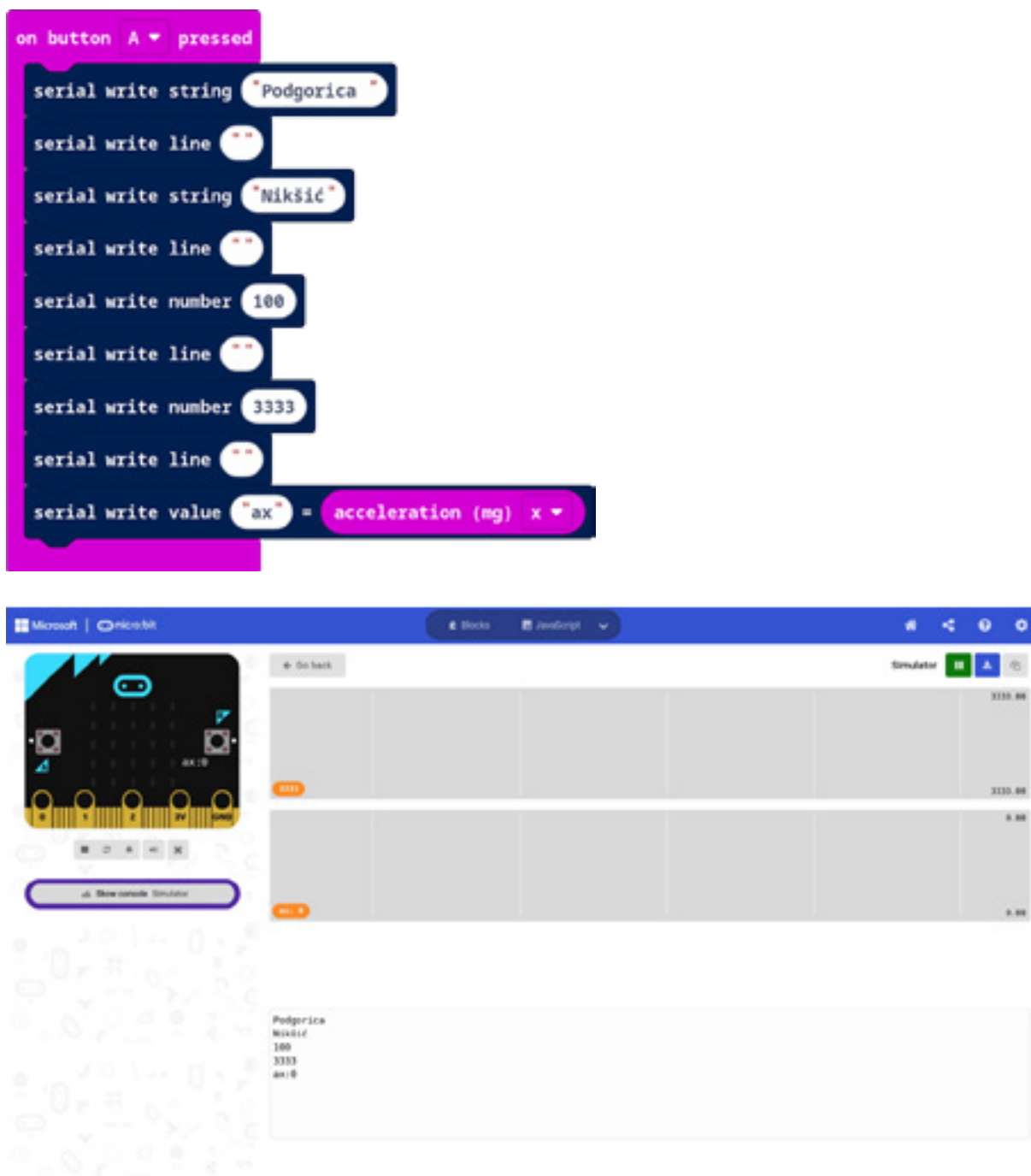
Pokreću program i prelaze na prikaz rezultata klikom na dugme „show console Simulator”. Komentarišu rezultate programa i činjenicu da serijsko slanje podataka spaja sve podatke u jedan. Slušaju objašnjenje nastavnika: ako podaci predstavljaju neprekidni tok podataka (engl. *stream*), prikazaće se i kao grafik i kao vrijednosti na konzoli. Ako je u pitanju tekst, prikazaće se samo na konzoli. Tekst tj. string se ne prikazuje na konzoli dok se ne kompletira cio red. Povratak na blokove se realizuje klikom na dugme *Go Back*.



Aktivnost 4

Učenici dodaju blokove *serial write line ...*, čime u konzolu dodaju novi red:

Diskutuju o prikazu koji dobijaju na ovaj način (svaki string je u posebnom redu konzole; brojevi se mogu posmatrati kao stream-ovi, pa ih DataView prikazuje i u posebnim redovima konzole i u dva grafika, po jedan za svaki broj):



The image shows a Scratch script for a micro:bit simulator. The script is triggered by 'on button A pressed' and contains the following blocks:

- `serial write string "Podgorica"`
- `serial write line ""`
- `serial write string "Nikšić"`
- `serial write line ""`
- `serial write number 100`
- `serial write line ""`
- `serial write number 3333`
- `serial write line ""`
- `serial write value "ax" = acceleration (mg) x`

The simulator interface shows the micro:bit hardware on the left and a data table on the right. The data table has two rows of data, each with two columns of numerical values (3333.00 and 9.00). Below the table, a console window displays the output of the script:

```
Podgorica
Nikšić
100
3333
ax:0
```

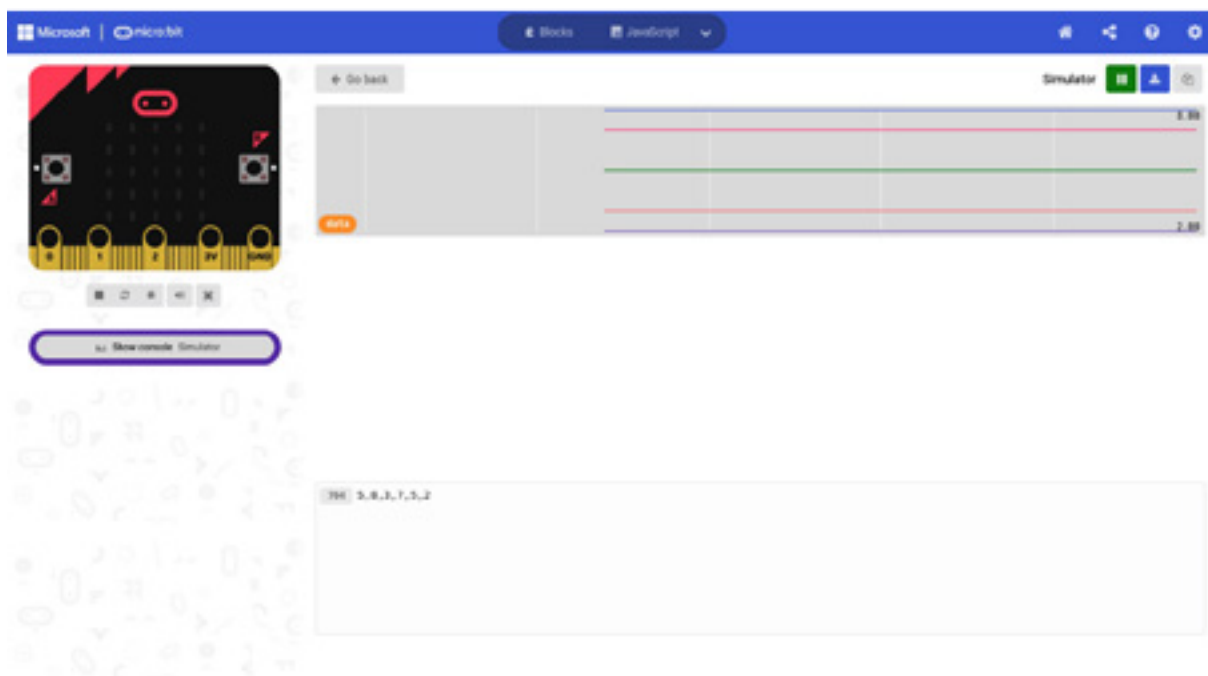
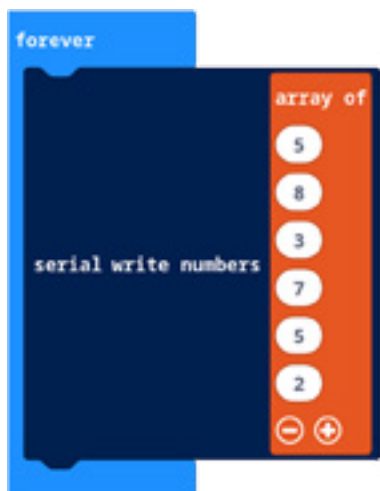
Aktivnost 5

Učenici dodaju blok *on button B pressed*, a zatim iz sekcije *Serial* dodaju 2 bloka *serial write line ...*, u kojima spajaju stringove primjenom bloka *join* iz sekcije *Text*. Prvi string se dobija blokom, spajanjem (join) stringova "Hello", " " i "World". Učenici slušaju objašnjenje da u drugom stringu između znaka navoda stoji simbol blanko.



Drugi string prikazuje ime i prezime učenika.

Učenici dodaju blok *serial write numbers ...* bloku *forever*, popunjavaju ga nizom [5, 8, 3, 7, 5, 2] i prate izvršavanje programa.

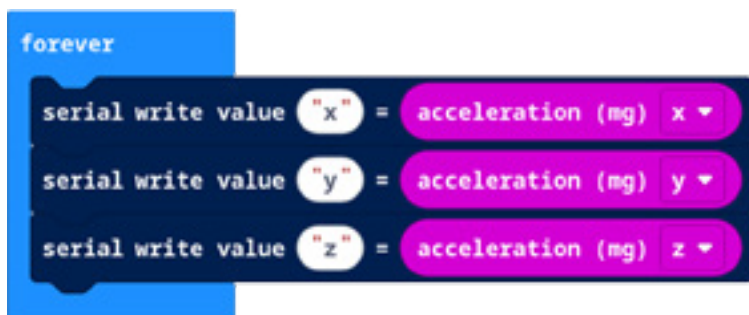


Za svaku vrijednost iz niza prikazuje se po jedan grafik.

Aktivnost 6

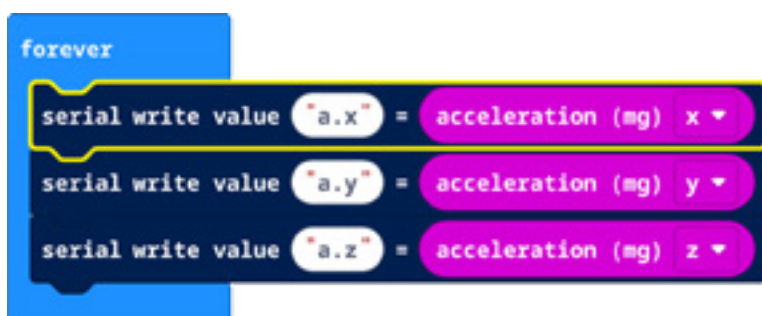
Učenici kreiraju novi projekat i bloku *forever* dodaju tri para *naziv – vrijednost*: (x, acceleration (mg) x), (y, acceleration (mg) y) i (z, acceleration (mg) z).

Izvršavaju program u simulatoru i simuliranjem pomjeranja uređaja mijenjaju se x, y i z komponenta ubrzanja, što utiče na svaku od tri krive.



Aktivnost 7

Učenici mijenjaju tri bloka u bloku *forever* tako da imena postaju "a.x", "a.y" i "a.z". Sada sva tri bloka imaju isto ime a, ali dodali su im „koordinate" x, y i z.



Izvršavaju program u simulatoru, ali se sada sve tri krive prikazuju na istoj slici. Učenici simuliranjem pomjeranja uređaja mijenjaju komponente x, y i z ubrzanja, što utiče na svaku od tri krive.



Aktivnost 8

Učenici izvršavaju operaciju privremenog zaustavljanja toka podataka.



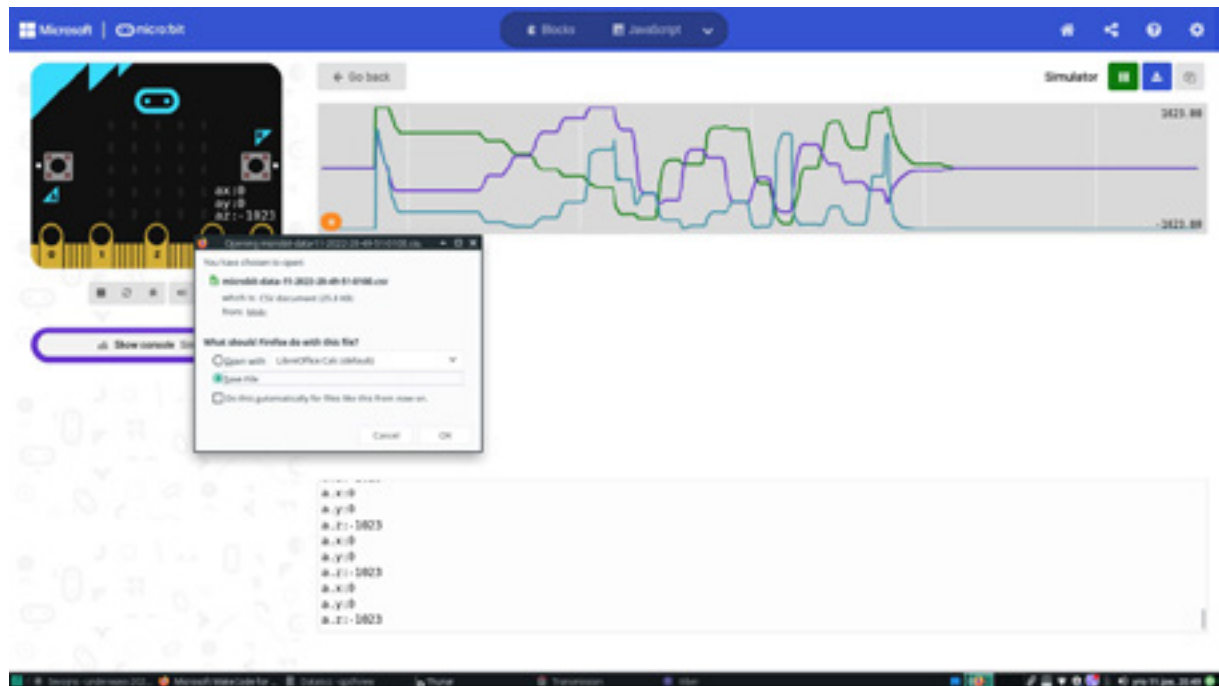
Klikom na prvo dugme, privremeno zaustavljaju tok podataka, pa bilo kakvo pomjeranje uređaja neće dovesti do promjena na grafiku.



Ponovnim klikom pokreću zaustavljeni tok podataka.

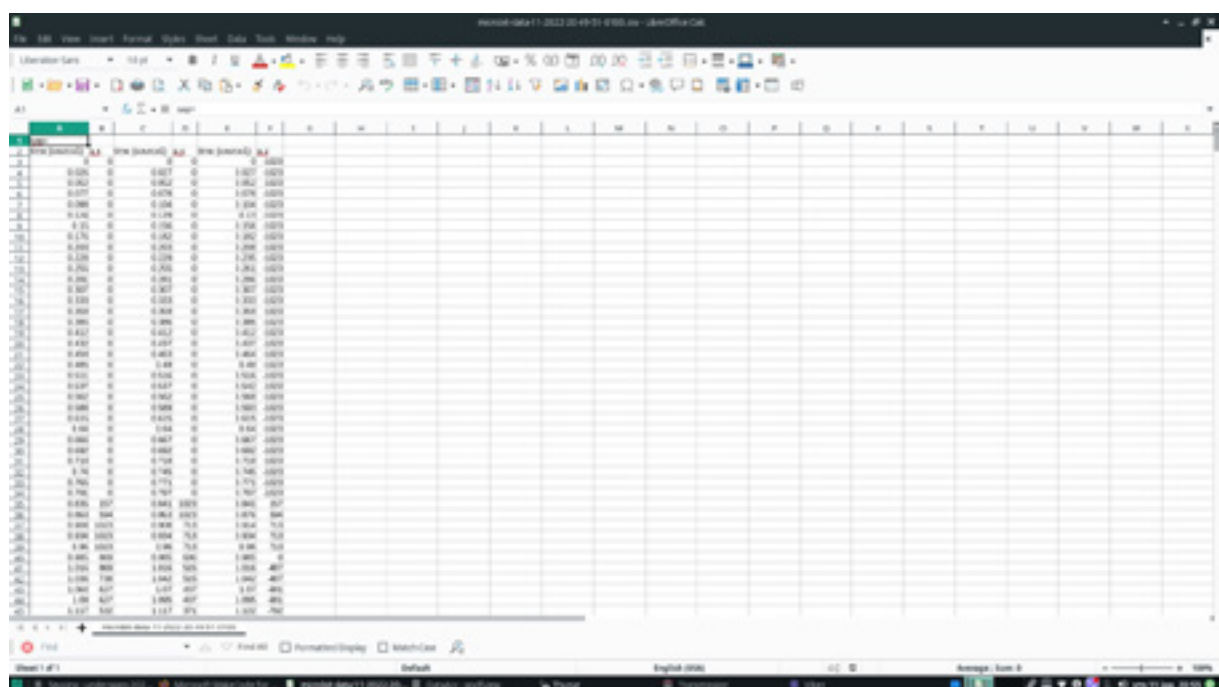
Aktivnost 9

Drugim dugmetom preuzimaju tzv. CSV fajl (CSV je skraćenica za *Comma Separated Values*). Ime fajla se automatski dodjeljuje od strane uređaja na osnovu datuma i vremena i ima sljedeći oblik: microbit-data-12-2022-19-25-02-0100.csv



Učenici otvaraju ovaj fajl nekim od programa za tabelarne proračune kao što su MS Excel, Libre Office Calc, Google Sheets, itd.

Diskutuju o kolonama koje su prikazane.



Aktivnost 10

Treće dugme učenici koriste da preuzmu fajl sa ekstenzijom „.txt”. Ime fajla se automatski dodjeljuje od strane uređaja na osnovu datuma i vremena i ima sljedeći oblik: microbit-console-12-2022-19-07-18-0100.txt

Taj fajl učenici otvaraju bilo kojim dostupnim editorom teksta i diskutuju o njegovom sadržaju.



```
932a.y:0
933a.z:-1023
934a.x:0
935a.y:0
936a.z:-1023
937a.x:0
938a.y:0
939a.z:-1023
940a.x:0
941a.y:0
942a.z:-1023
943a.x:0
944a.y:0
945a.z:-1023
946a.x:0
947a.y:0
948a.z:-1023
949a.x:0
950a.y:0
```

Programe možete pogledati kada skenirate QR kôdove ili na linkovima:

https://makecode.microbit.org/_8cy0oHMw96RJ

https://makecode.microbit.org/_KbY8PkEu4Yfb





PRIMJERI PRAĆENJA PROCJENJIVANJA

Akcenat obrazovno-vaspitnih ishoda koji su obuhvaćeni ovim priručnikom je na praktičnom znanju iz oblasti programiranja i elektronike, pa je suština aktivnosti u fazi provjere i ocjenjivanja znanja na radu sa praktičnim vježbama, a ne samo na teorijskim znanjima iz oblasti ICT-a. Brojčanom ocjenom (od 1 do 5) treba vrednovati praktične aktivnosti, ali i pokazano znanje i razumijevanje. Takođe, treba vrednovati kreativnost, originalnost rješenja, radno-tehničku spretnost, uvažavajući interesovanja, zalaganja i individualne sposobnosti učenika.

Primjeri kriterijuma ocjenjivanja po razredima:

VIII RAZRED		
NIVO POSTIGNUĆA		
Osnovni nivo	Srednji nivo	Viši nivo
<ul style="list-style-type: none"> - definiše pojam niza i prepoznaje ga u svakodnevnom životu; - demonstrira sortiranje niza jednim od algoritama, uz pomoć nastavnika, na datom primjeru; - implementira programe, uz pomoć nastavnika, koji definišu niz i pristupaju ili mijenjaju elemente niza; - implementira programe, uz pomoć nastavnika, koji razmjenjuju poruke između 2 Micro:bit uređaja; - uz pomoć nastavnika realizuje prenos podataka sa uređaja Micro:bit u tekstualni ili CSV fajl; - komunikacija i saradnja sa ostalim učenicima nije razvijena. 	<ul style="list-style-type: none"> - razlikuje nizove brojeva, stringova i objekata; - demonstrira sortiranje niza primjenom dva algoritma na datom primjeru; - prilikom izrade programa traži povremeno pomoć nastavnika ili boljih učenika; - prezentuje programe i zadatke uz povremene greške; - samostalno realizuje prenos podataka sa uređaja Micro:bit u tekstualni ili CSV fajl; - povremeno se uključuje i komentariše; - komunicira i sarađuje samo sa manjim brojem učenika. 	<ul style="list-style-type: none"> - demonstrira sortiranje niza primjenom sva tri algoritma na datom primjeru; - samostalno implementira programe; - pronalaži i otklanja greške u programu; - odlično se snalazi u prezentovanju zadataka; - samostalno realizuje prenos podataka sa uređaja Micro:bit u tekstualni i CSV fajl; - aktivno učestvuje u razgovoru iznoseći jasno svoje mišljenje i obrazloženje, daje svoje predloge za neka rješenja.



PROJEKTNASTAVA

Primjer projekta za simulaciju slučajnih događaja opisan je u radnom listu „Simulacija slučajnih događaja“ u dodatku 2.

Primjeri projekata koji koriste nizove:

<https://makecode.microbit.org/courses/csintro/arrays/project>

Primjeri projekata koji koriste Radio:

https://www.electfreaks.com/learn-en/microbitKit/Tinker_Kit/tinker_kit_case_12.html
<https://tinkercademy.com/tutorials/make-a-microbit-car/>
<https://tinkercademy.com/tutorials/remote-control-everything/>
<https://littlebirdelectronics.com.au/guides/86/bluetooth-with-micro-bit>
<https://www.instructables.com/Radio-Signals-on-Microbit/>
<https://micromag.cc/micro-bit-powered-fm-radio/>
<https://aca.edu.au/resources/python-microbit-radio/offline-78-microbit-radio.pdf>

Primjeri projekata koji koriste datoteke:

<https://microbit.org/lessons/data-handling-unit-summary/>
<http://multiwingspan.co.uk/micro.php?page=files>
<https://docs.microsoft.com/en-us/microsoft-365/education/data-streamer/using-microbit-and-makecode>
<https://blog.penjee.com/how-to-create-read-and-write-a-file-with-microbit/>
<https://microbit-micropython.readthedocs.io/en/latest/filesystem.html>
<https://techtutorialsx.com/2019/02/10/microbit-micropython-creating-a-file-in-the-filesystem/>
https://www.youtube.com/watch?v=_bQtl_Lwpd4
<https://xinabox.cc/pages/data-logging-with-the-bbc-micro-bit>
<https://www.stem.org.uk/system/files/community-resources/2017/02/First%20steps%20with%20microbits%20for%20data-logging%20and%20modelling.pdf>
<https://digitalmaestro.org/articles/microbit-progress-bar-for-data-collection-projects>
<https://digitalmaestro.org/articles/microbit-temperature-sensor-data-collection>
<https://www.youtube.com/watch?v=-WrPJ4e4Jf0>
<https://learn.adafruit.com/data-logging-and-file-storage-in-makecode>
<https://www.hackster.io/HackingSTEM/visualize-data-in-excel-with-micro-bit-and-makecode-be21fb>
<https://support.microbit.org/support/solutions/articles/19000016212-storing-and-retrieving-values-in-non-volatile-data>



OPREMA KOJA JE KORIŠĆENA ZA REALIZACIJU ISHODA

- Micro:bit;
- mikro USB kabl;
- baterija sa 2 x 1,5V baterije.



KLJUČNI POJMOVI PREMA REDOSLJEDU POJAVLJIVANJA

Promjenljive ili varijable predstavljaju podatke kojima se upravlja u programima. Promjenljive ili varijable mogu mijenjati vrijednost u toku izvršavanja programa – otuda potiče riječ promjenljiva ili na engleskom *variable*. U najprostijem smislu, to su imenovani dijelovi memorije kojima se pristupa u programu. Imena promjenljivih mogu sadržati cifre, slova i donju crticu, a moraju početi slovom ili donjom crticom.

Euklidov algoritam se ukratko i ne previše precizno može opisati ovako: veći broj se umanjuje za vrijednost manjeg, novodobijene vrijednosti se porede i postupak se ponavlja sve dok se brojevi ne izjednače.

Naredba *if* je naredba kontrole toka programa i omogućava izvršavanje određenog dijela programa u zavisnosti od toga da li je navedeni uslov ispunjen ili ne. Ukoliko je uslov ispunjen, tj. ukoliko je vrijednost logičkog izraza True, onda se izvršava jedna naredba ili blok naredbi navedenih unutar naredbe *if*. Ukoliko uslov nije ispunjen, kôd programa unutar naredbe *if* se neće izvršiti.

While petlja je dio kôda koji se izvršava dokle kod je uslov te petlje zadovoljen. Dakle, kada dođemo do petlje u kodu, računar provjerava da li je uslov ispunjen, ako jeste, on izvršava komande unutar petlje. Nakon što ih izvrši, vraća se ponovo do uslova, ako je on i dalje ispunjen, komande unutar petlje se ponovo izvršavaju. Sve se to ponavlja dokle god uslov ne postane netačan.

Operatori poređenja, kako im samo ime kaže, služe za upoređivanje dvije vrijednosti. To su sljedeći operatori:

1. **== (jednako)** je operator koji provjerava da li su vrijednosti sa lijeve i desne strane operatora jednake;
2. **!= (različito)** je operator koji provjerava da li su vrijednosti sa lijeve i desne strane operatora različite;
3. **< (manje od)** je operator koji služi za provjeru da li je vrijednost na lijevoj strani manja od vrijednosti na desnoj strani;
4. **> (veće od)** je operator koji služi za provjeru da li je vrijednost na lijevoj strani veća od vrijednosti na desnoj strani;
5. **<= (manje ili jednako)** je operator koji služi za provjeru da li je vrijednost na lijevoj strani manja od vrijednosti na desnoj strani ili jednaka sa njom;
6. **>= (veće ili jednako)** je operator koji služi za provjeru da li je vrijednost na lijevoj strani veća od vrijednosti na desnoj strani ili jednaka sa njom.

Rezultat primjene ovih operatora može imati vrijednost „tačno“ (True) ili „netačno“ (False).

Logički operatori omogućavaju da se usmjerava tok izvršavanja programa. Postoje 3 logička operatora i to su operatori: and (i), or (ili) i not (ne).

Operator and je logički operator čiji rezultat izvršavanja ima logičku vrijednost True samo ako oba izraza sa lijeve i desne strane operatora and imaju logičku vrijednost True. Inače, rezultat izvršavanja ima logičku vrijednost False.

Operator or je logički operator čiji rezultat izvršavanja ima logičku vrijednost False samo ako oba izraza sa lijeve i desne strane operatora and imaju logičku vrijednost False. Inače, rezultat izvršavanja ima logičku vrijednost True.

Operator not je logički operator koji se primjenjuje nad samo jednim izrazom. Ukoliko taj izraz ima logičku vrijednost True, primjenom operatora not, logička vrijednost izraza postaje False. U suprotnom, ako je logička vrijednost izraza False, primjenom operatora not, logička vrijednost izraza postaje True. Operacija koja se izvršava primjenom operatora not zove se negacija.

While – slobodni prevod engleske reči while mogao bi u programiranju da se interpretira kao: „Radi naredni kod dok je ispunjen uslov...“. Dakle, *while* petlja se izvršava više puta sve dok je uslov True, a else grana se izvršava samo jednom, onda kada uslov postane False.

Linijski program / linijski algoritam je program kod kojeg za svaki korak imamo jednu naredbu i svaki korak se izvršava jedanput. Linijski algoritmi su osnovni i predstavljaju najjednostavnije programe. To su programi koji imaju ulaz (unos podataka), izlaz (ispis podataka) i između ta dva koraka obradu podataka.

String predstavlja niz karaktera, predstavljenih između znakova navoda. Stringovi imaju veliku primjenu u programiranju. Za Micro:bit uređaj podržani su samo ASCII kodovi znakova od 32 do 126. U te znake spadaju slova, cifre, interpunkcijski znaci i nekoliko simbola. Svi ostali znaci pojavljuju se kao ? na ekranu Micro:bit-a.

CSV fajl – fajl koji sadrži podatke razdvojene simbolom zarez, engl. „Comma Separated Values”.

Farenhajt – mjerna jedinica za temperaturu, u upotrebi u SAD.

Collatz Conjecture (Kolazova pretpostavka) – nedokazan matematički problem koji ukazuje na pretpostavku da se svaki nasumičan broj primjenom 2 pravila, u zavisnosti od toga da li je broj paran ili neparan, svede na petlju 4,2,1.

Sintaksa programskog jezika – tekstualni izraz programa.

Uzemljenje – sistem za odvođenje nepoželjne električne struje u tlo/zemljište.

GND – u elektronici, internacionalna skraćenica za uzemljenje.

Relej – električno napajan prekidač strujnog kola.

Propeler – element s glavom u sredini iz koje se protežu oštrice postavljene tako da svaka čini dio jedne spiralne površine.

Proto-ploča – konstruktivna osnova za izradu elektronskih prototipova i strujnih kola.

Niz – struktura u programskom jeziku koja sadrži više elemenata istog tipa.

Indeks niza – svakom elementu pridružen je jedan broj koji se naziva indeks elementa i koji označava poziciju elementa u nizu.

Dužina niza – broj elemenata u nizu.

Sortiranje niza – postupak rasporeda elemenata niza tako da budu raspoređeni u nekom poretku.

Bubble sort – algoritam sortiranja u kojem se upoređuju susjedni elementi niza i ako nijesu u odgovarajućem poretku, zamjenjuju im se mjesta. Ovaj postupak se ponavlja sve dok ima zamjene mjesta u nizu. Sortiranje se obično obavlja u samom nizu (engl. *in-place*), bez uvođenja pomoćnog niza.

Selection sort – metod sortiranja koji se zasniva na podjeli niza na sortirani i nesortirani dio, na pronalaženju najmanjeg elementa u nesortiranom dijelu niza i zamjeni mjesta elemenata niza.

Insertion sort – algoritam uklanja elemente jedan po jedan iz nesortiranog niza i umeće ih u sortirani dio niza na odgovarajuću poziciju. Sortiranje se obično obavlja u samom nizu (engl. *in-place*), bez uvođenja pomoćnog niza. Elementi se uklanjaju sa početka ili kraja niza, čime se stvara prostor za sortirani dio niza.

Par „ime – vrijednost” – zadavanje imena i odgovarajuće vrijednosti u programima, npr. za slanje poruka radio-vezom.

LITERATURA

1. Gareth Halfacree, "The Official BBC micro:bit User Guide", Wiley, 2017, ISBN: 978-1-119-38673-5
2. Wolfram Donat, "Getting Started with the micro:bit - Coding and Making with the BBC's Open Development Board", MakerMedia, 2017, ISBN: 978-1-680-45302-7
3. Pradeeka Seneviratne, "Beginning BBC micro:bit: A Practical Introduction to micro:bit Development", Apress, 2018, ISBN-13 (electronic): 978-1-4842-3360-3
4. Simon Monk, "Programming the BBC micro:bit – Getting Started with MicroPython", Mc Graw Hill, 2018, ISBN: 978-1-26-011759-2
5. Pradeeka Seneviratne, "BBC micro:bit Recipes: Learn Programming with Microsoft MakeCode Blocks", Apress, 2019, ISBN-13 (electronic): 978-1-4842-4913-0
6. Lorraine Underwood, "Save the World with the Code: 20 Fun Projects for All Ages Using Raspberry PI, micro:bit, and Circuit Playground Express", Mc Graw Hill, 2020, ISBN: 978-1-26-045760-5
7. Philip Meitiner, Pradeeka Seneviratne, "Beginning Data Science, IoT, and AI on Single Board Computers: Core Skills and Real-World Application with the BBC micro:bit and XinaBox", Apress, 2020, ISBN-13 (electronic): 978-1-4842-5766-1
8. Online priručnik „Programiranje Micro:bit micro:bit uređaja u MejkKodu – priručnik za peti razred”, <https://petlja.org/biblioteka/r/kursevi/microbitbc>
9. kitronik.co.uk
10. <https://microbit.org/lessons/>
11. Intro to CS with the micro:bit <https://makecode.microbit.org/courses/csintro>

DODATAK 1 – PRILOG ZA ISHOD „SORTIRANJE NIZA SELEKCIJOM”

Lista aktivnosti za ishod „Sortiranje selekcijom”

Uvod

Postoji više načina sortiranja niza. Naučićeš jedan jednostavan metod sortiranja, koji se naziva sortiranje selekcijom. Metod se zasniva na podjeli niza na sortirani i nesortirani dio, na pronalaženju najmanjeg elementa u nesortiranom dijelu niza i zamjeni mjesta elemenata niza. Algoritam za ovaj metod sortiranja je sljedeći:

1. Odredi najmanji elemenat u nizu. Zamjena mjesta se može uraditi na dva načina:
 - **direktno**, tako što pronađemo poziciju najmanjeg elementa i direktno izvršimo zamjenu;
 - **indirektno**, tako što redom posmatramo elemente desno od pozicije 0 i svaki put kada nađemo na manji element vršimo zamjenu; kada prođemo kroz sve elemente do kraja niza, na poziciji 0 biće najmanji element.
2. Odredi drugi najmanji elemenat u nizu. Zamjena se može izvršiti na dva načina, kako je opisano u koraku 1, samo što sada posmatramo elemente desno od pozicije 1.

3. Odredi treći najmanji elemenat u nizu. Zamijeni mjesta trećem elementu niza i trećem najmanjem elementu niza.
4. Ponavljaj postupak određivanja sljedećeg najmanjeg elementa i zamjene mjesta, sve dok niz ne bude uređen.

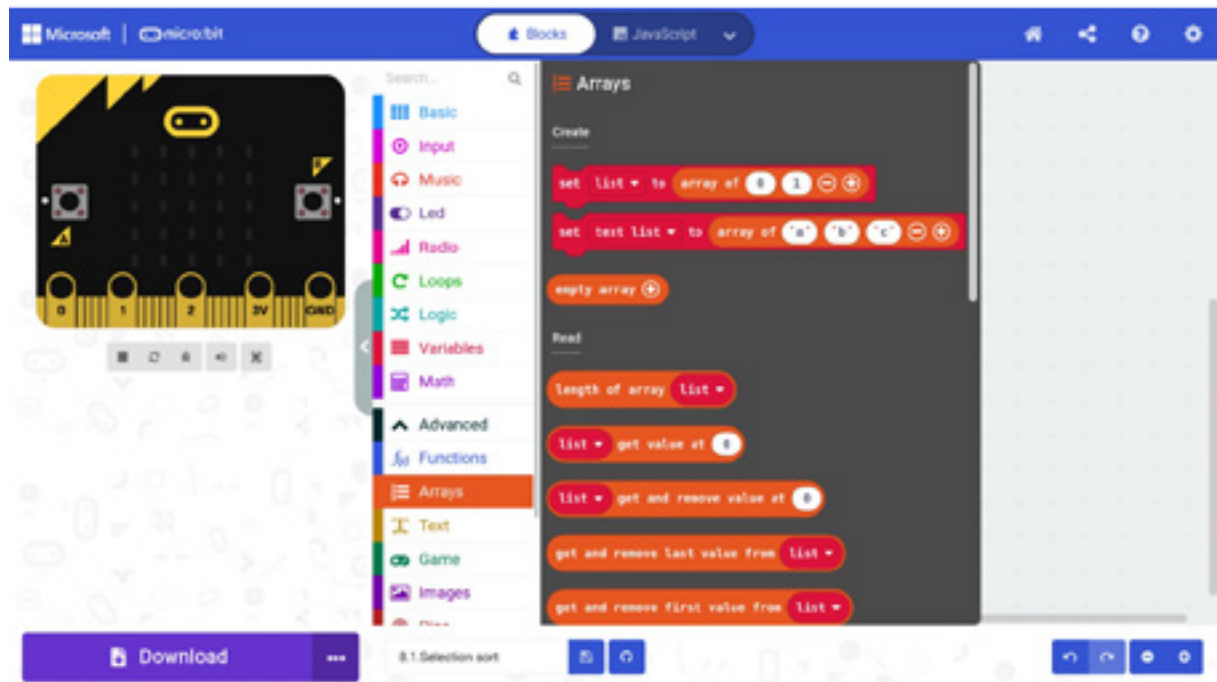
Primjer: Žuta boja označava poziciju na kojoj treba da bude sljedeći najmanji element.

Niz	Najmanji element	Niz poslije zamjene mjesta
[12, 8, 23, 6, 34, 22, 45, 9]	6	[6, 8, 23, 12, 34, 22, 45, 9]
[6, 8, 23, 12, 34, 22, 45, 9]	8	[6, 8, 23, 12, 34, 22, 45, 9]
[6, 8, 23, 12, 34, 22, 45, 9]	9	[6, 8, 9, 12, 34, 22, 45, 23]
[6, 8, 9, 12, 34, 22, 45, 23]	12	[6, 8, 9, 12, 34, 22, 45, 23]
[6, 8, 9, 12, 34, 22, 45, 23]	22	[6, 8, 9, 12, 22, 34, 45, 23]
[6, 8, 9, 12, 22, 34, 45, 23]	23	[6, 8, 9, 12, 22, 23, 45, 34]
[6, 8, 9, 12, 22, 23, 45, 34]	34	[6, 8, 9, 12, 22, 23, 34, 45]
[6, 8, 9, 12, 22, 23, 34, 45]	45	[6, 8, 9, 12, 22, 23, 34, 45]
[6, 8, 9, 12, 22, 23, 34, 45]		

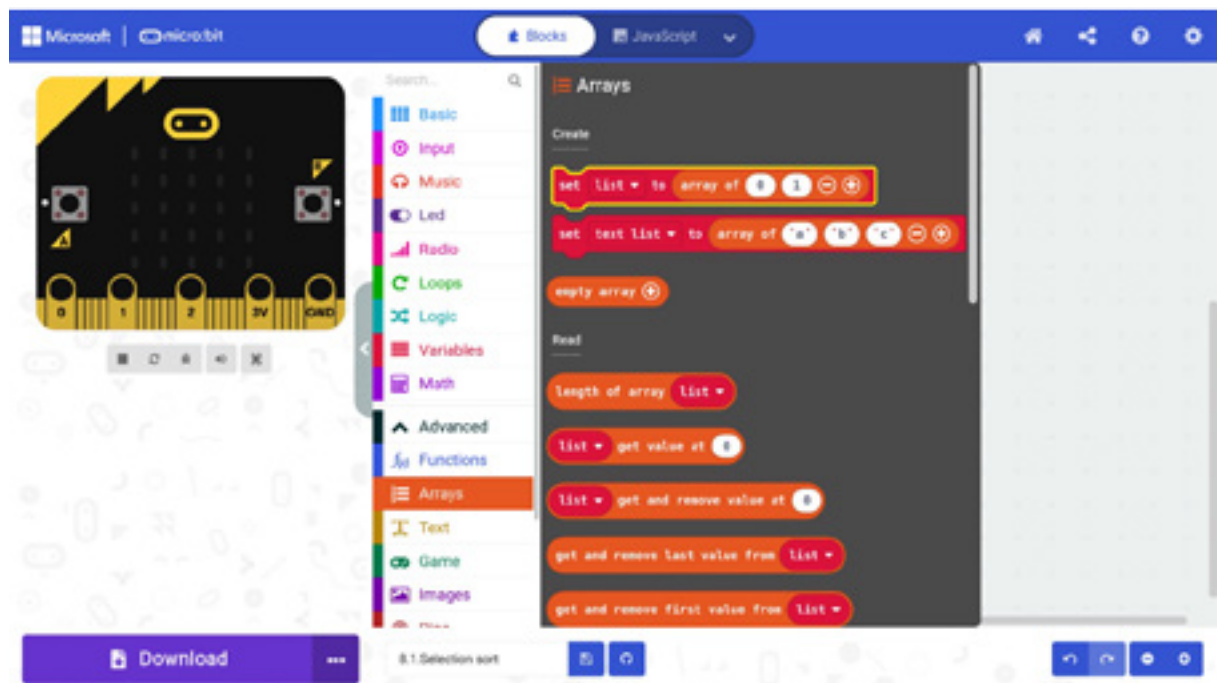
Primjer:

Sortirani dio niza	Nesortirani dio niza	Najmanji element u nesortiranom dijelu
[]	[12, 8, 23, 6, 34, 22, 45, 9]	6
[6]	[12, 8, 23, 34, 22, 45, 9]	8
[6, 8]	[12, 23, 34, 22, 45, 9]	9
[6, 8, 9]	[12, 23, 34, 22, 45]	12
[6, 8, 9, 12]	[23, 34, 22, 45]	22
[6, 8, 9, 12, 22]	[23, 34, 45]	23
[6, 8, 9, 12, 22, 23]	[34, 45]	34
[6, 8, 9, 12, 22, 23, 34]	[45]	45
[6, 8, 9, 12, 22, 23, 34, 45]	[]	

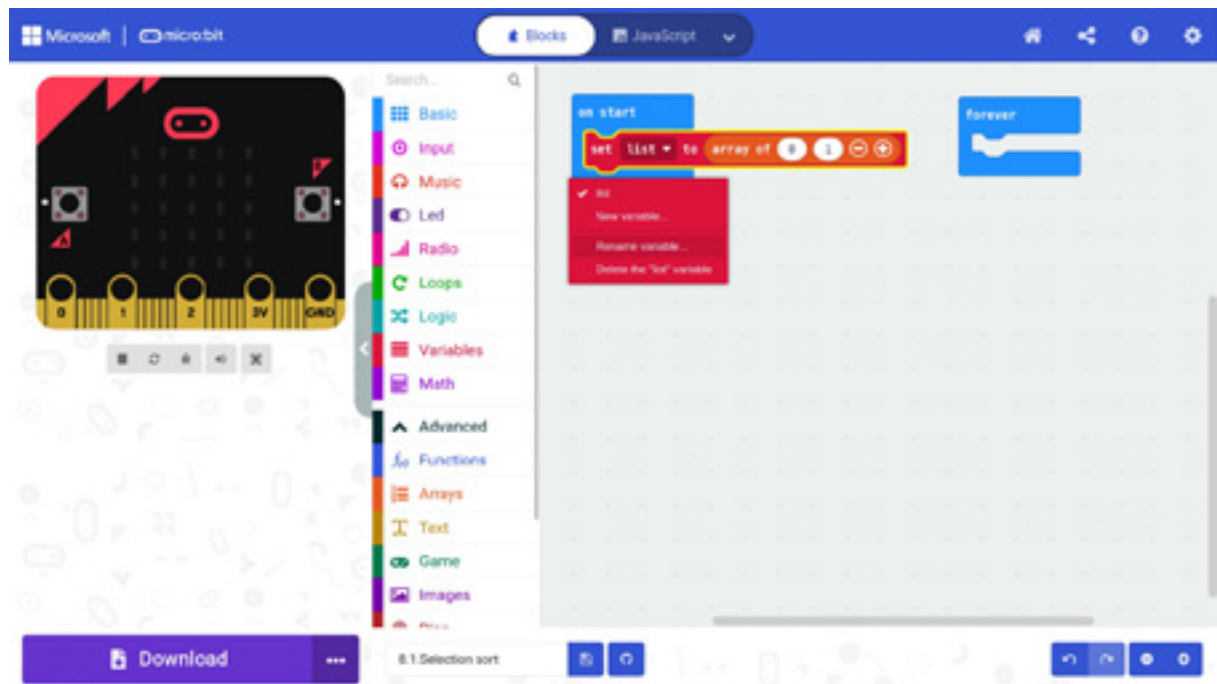
1. Otvori sekciju za nizove (*Advanced/Arrays*).



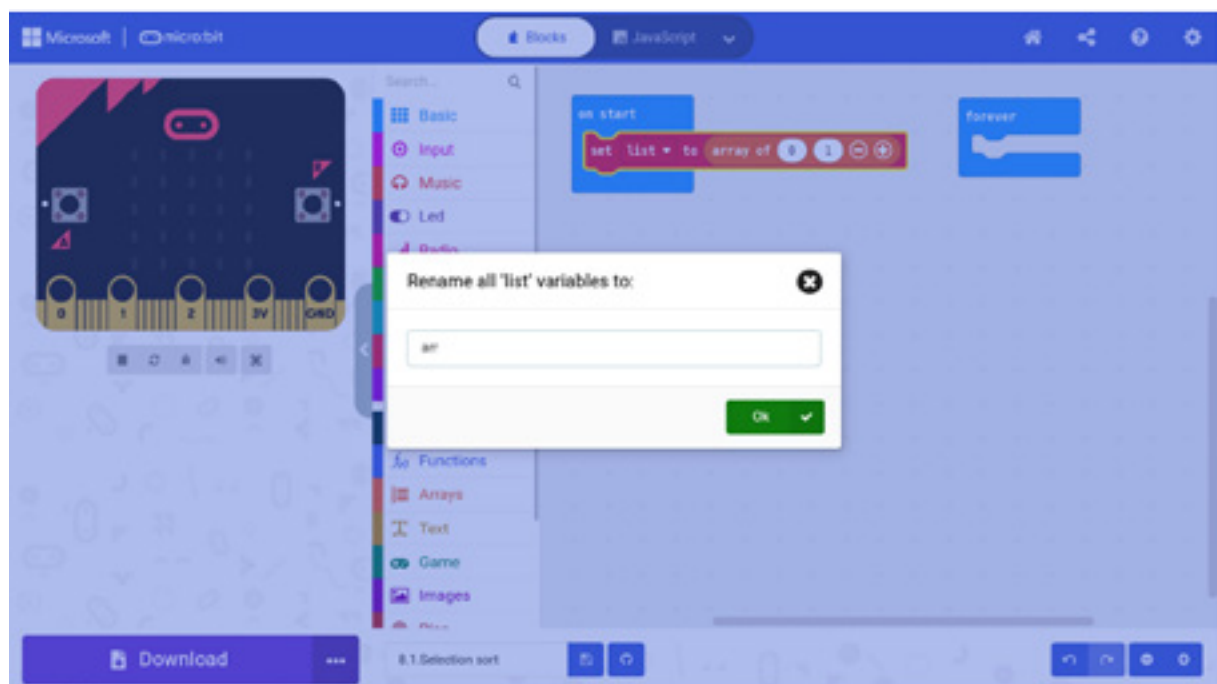
2. Odaberi blok *set list to array...*



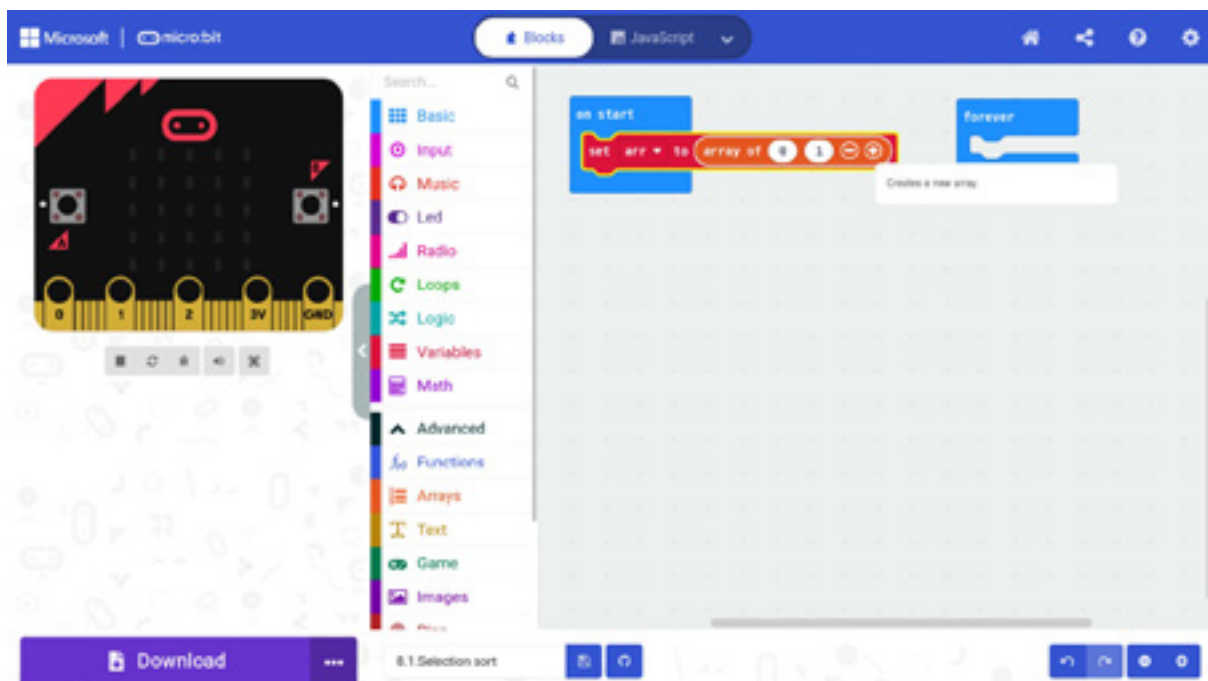
3. U blok *on start* dodaj blok *set list to array...* i promijeni ime niza.



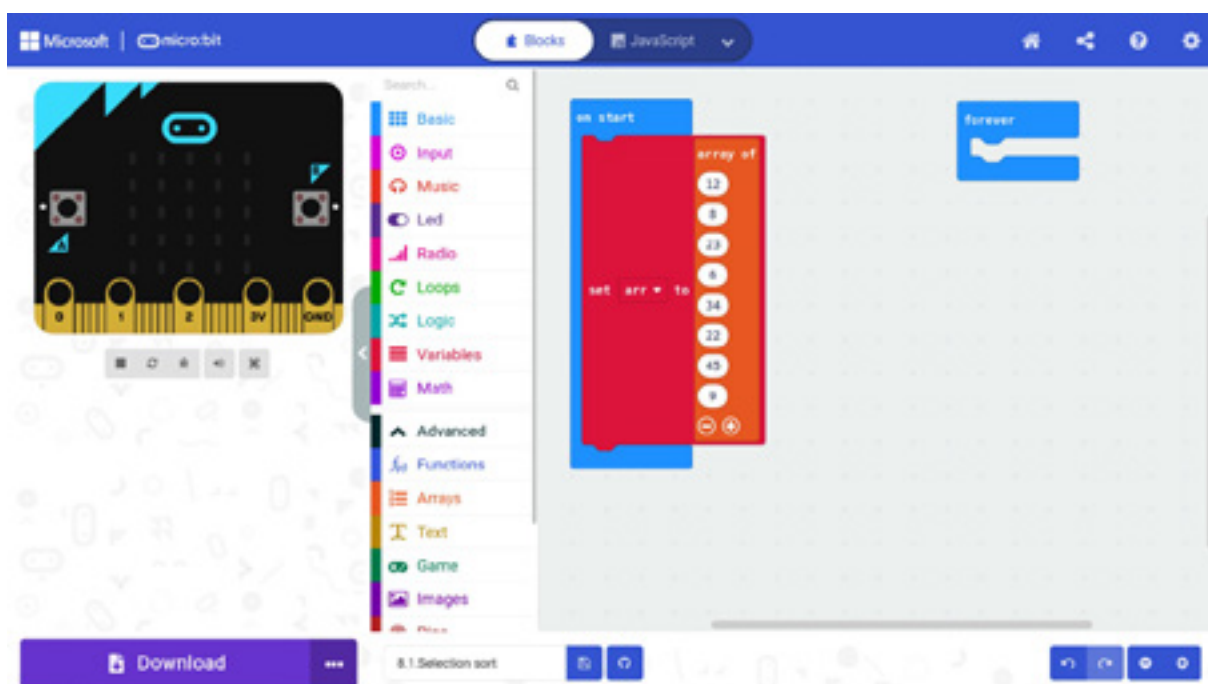
4. Novo ime niza je *arr*.



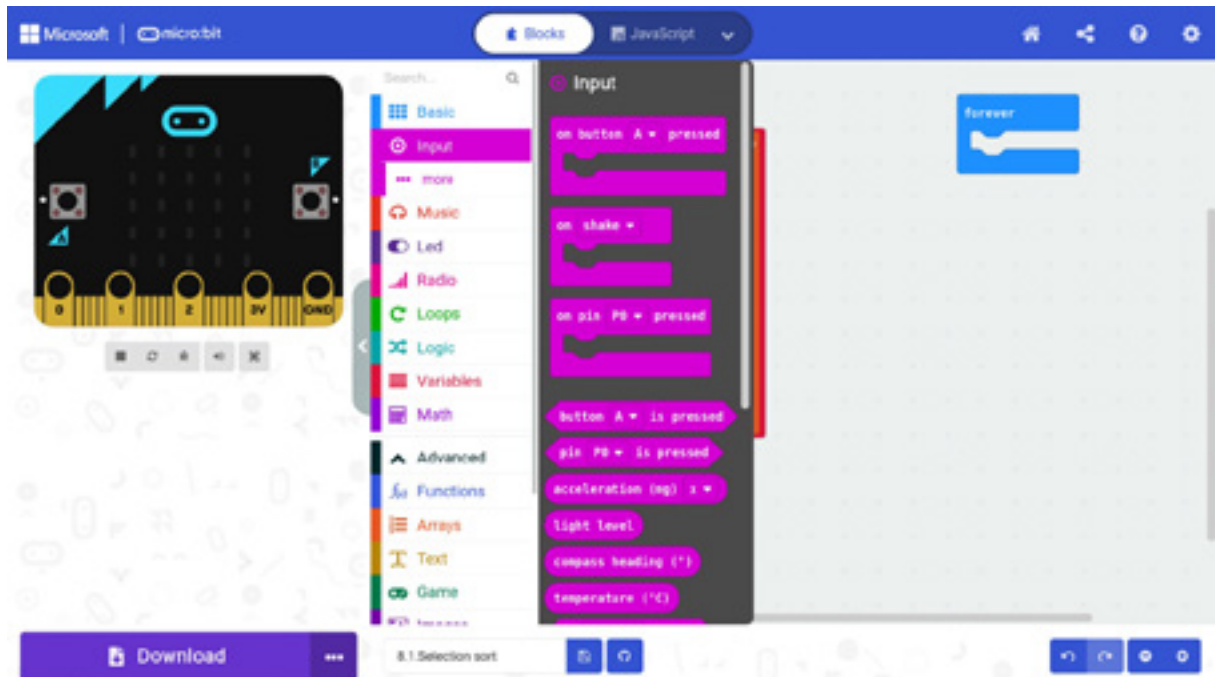
5. Dodaj nove elemente u niz.



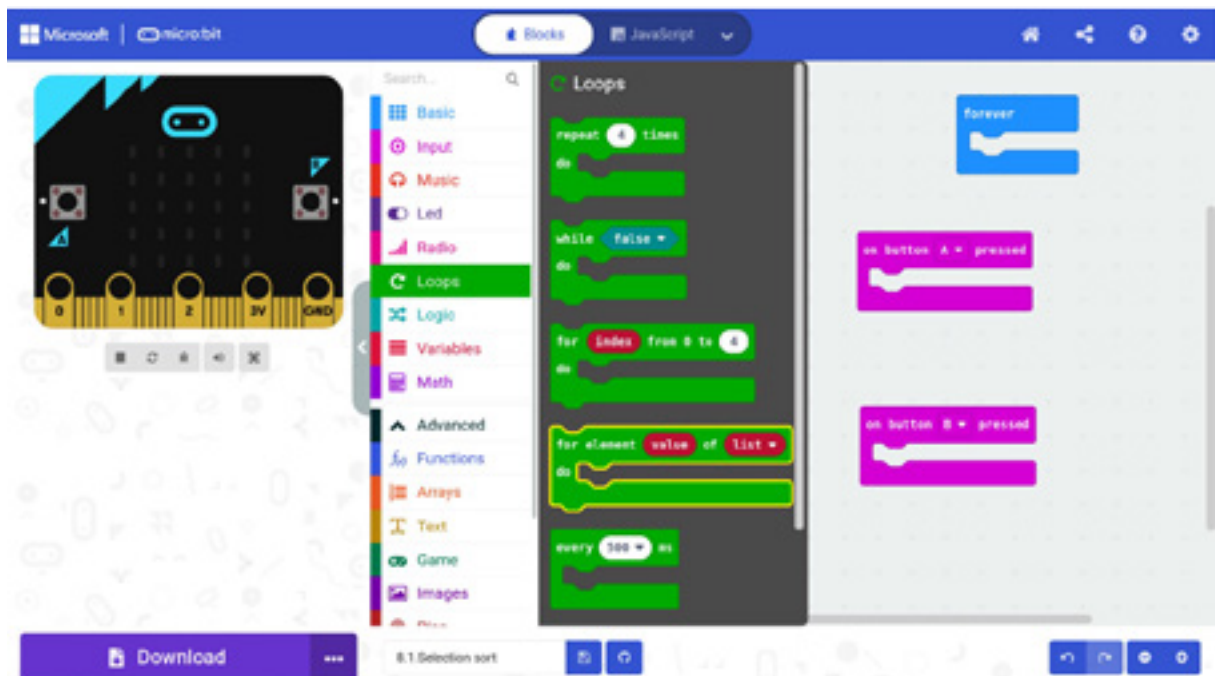
6. Niz *arr* sadrži 8 elemenata, koje zapisujemo na sljedeći način: [12, 8, 23, 6, 34, 22, 45, 9]. Ne zaboravi da će indeksi niza *arr* imati vrijednosti 0, 1, 2, 3, 4, 5, 6 i 7. Na primjer, pišemo *arr*[1] = 8 ili *arr*[6] = 45.



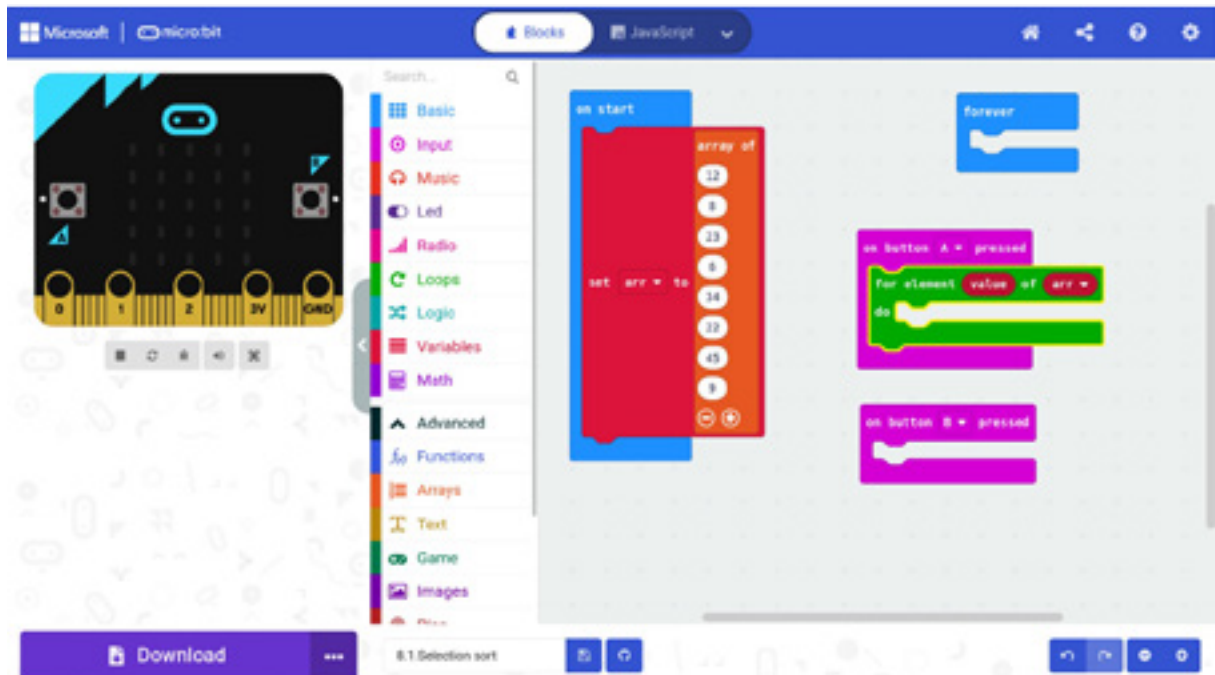
- Pritiskom na dugme A, štampaćeš elemente niza *arr*, a pritiskom na dugme B, sortiraćeš niz *arr* u rastući.



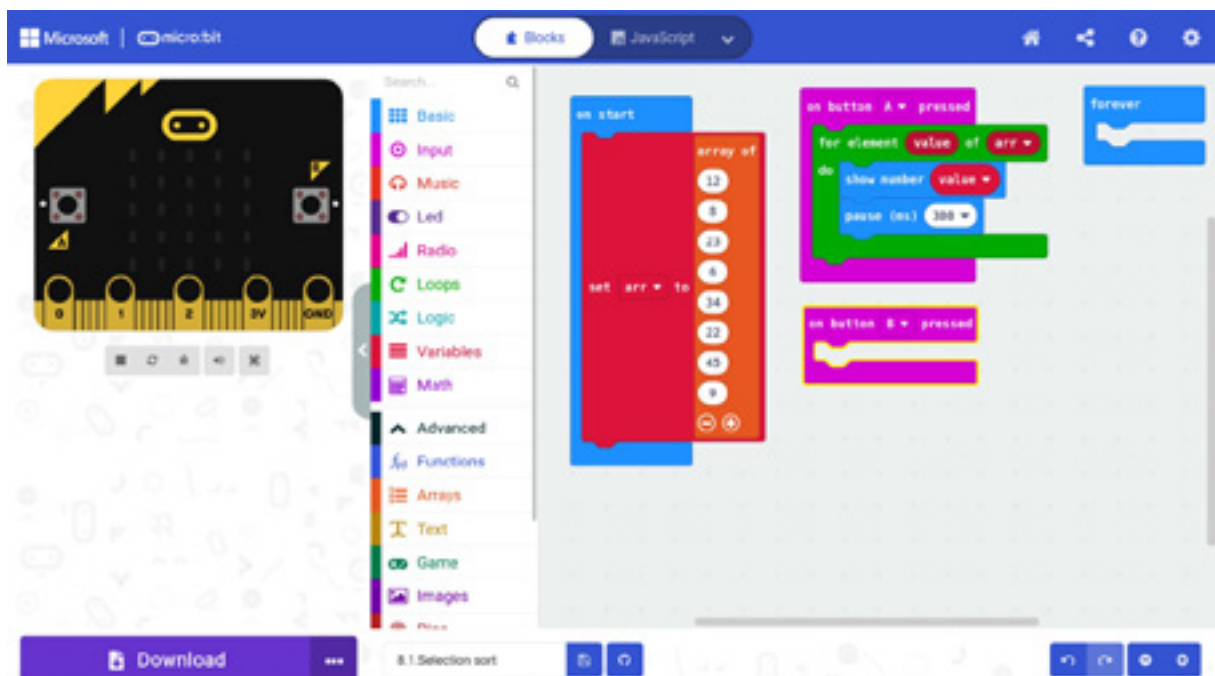
- Dodaj blokove *on button A pressed* i *on button B pressed*.



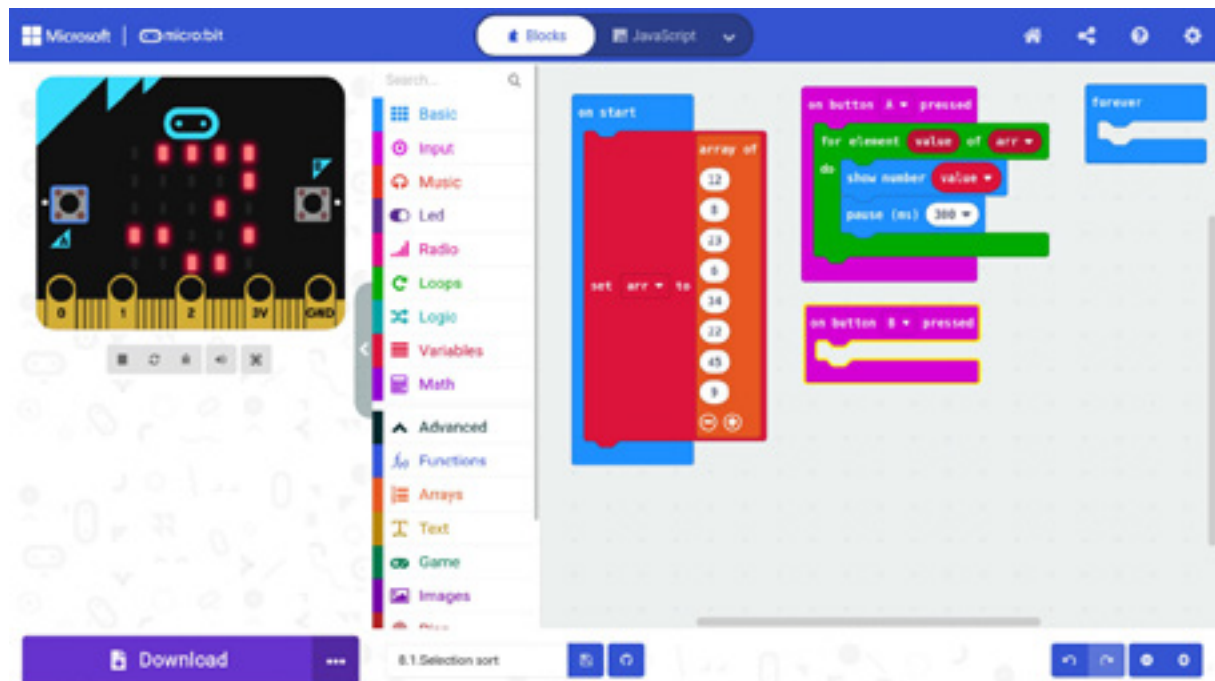
- U blok *on button A pressed* dodaj blok *for element value of arr*. Ne zaboravi da izabereš niz *arr*.



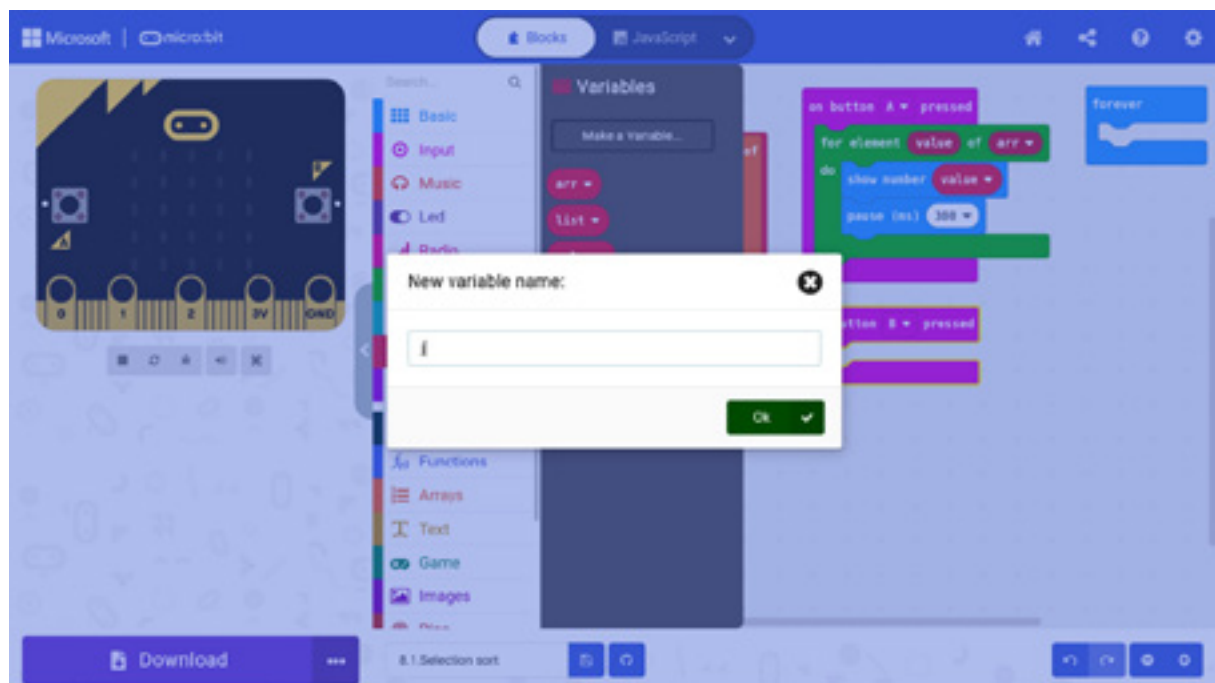
- U ciklus *for* dodaj i prikaži element niza (tj. *value*) i zatim pauziraj 300 ms.



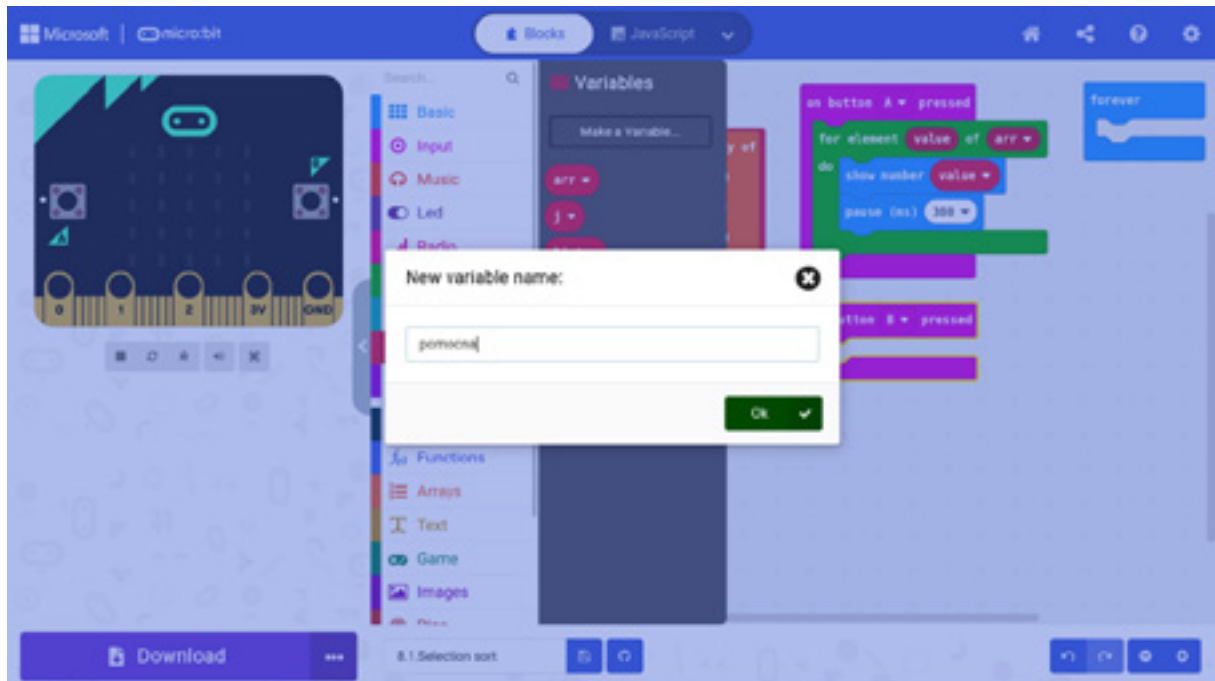
11. Pokrenite program u simulatoru. Na ekranu će se redom iscrtati svi elementi niza *arr*.



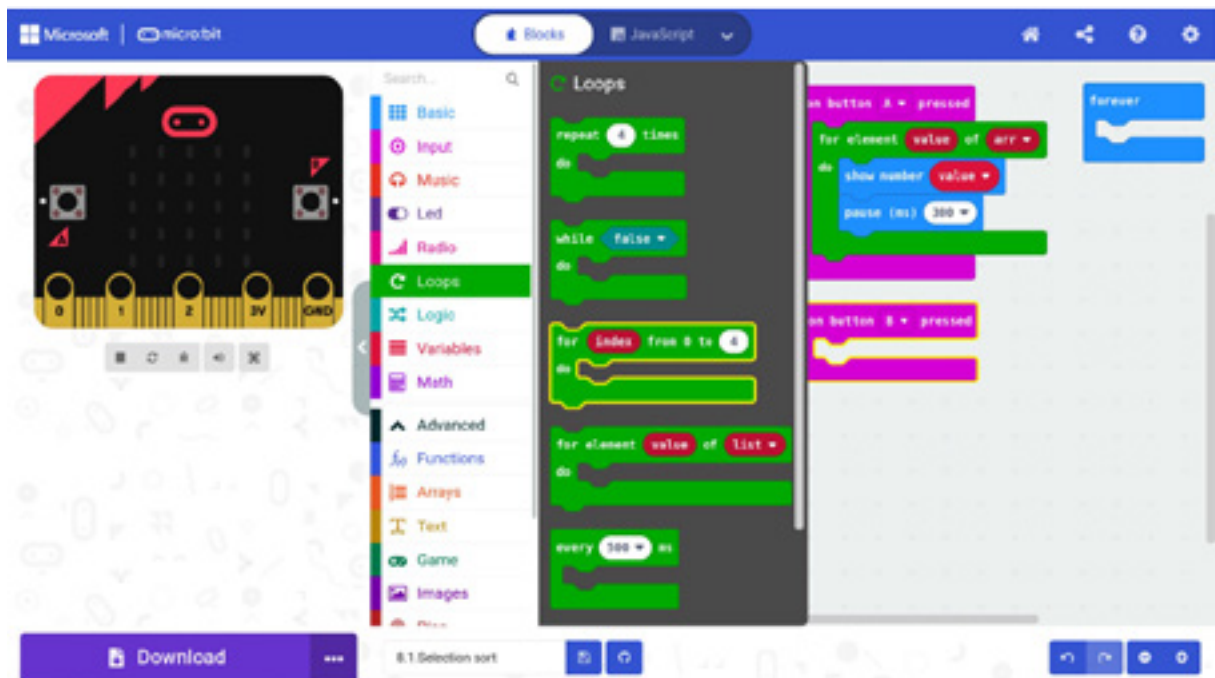
12. Za sortiranje niza potrebno je da definišeš novu promjenljivu „j“, koja će imati ulogu indeksa niza.



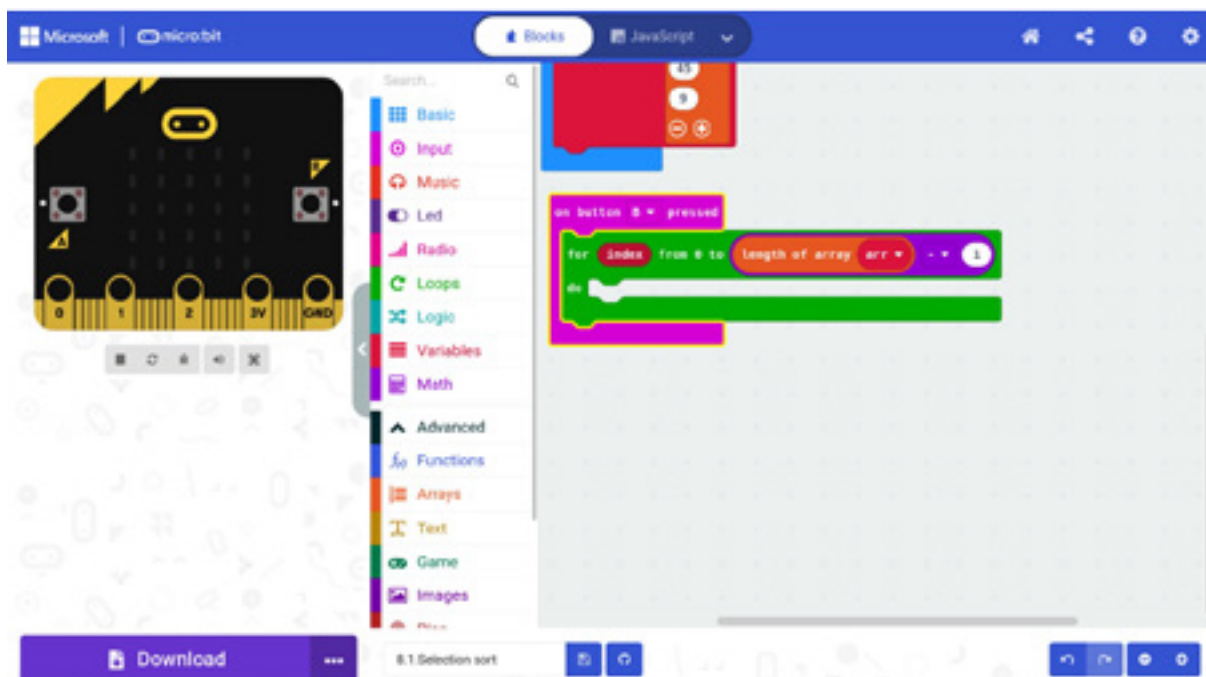
13. Definiši i promjenljivu „pomocna”, čija će uloga biti da nam pomogne prilikom zamjene mjesta elemenata niza.



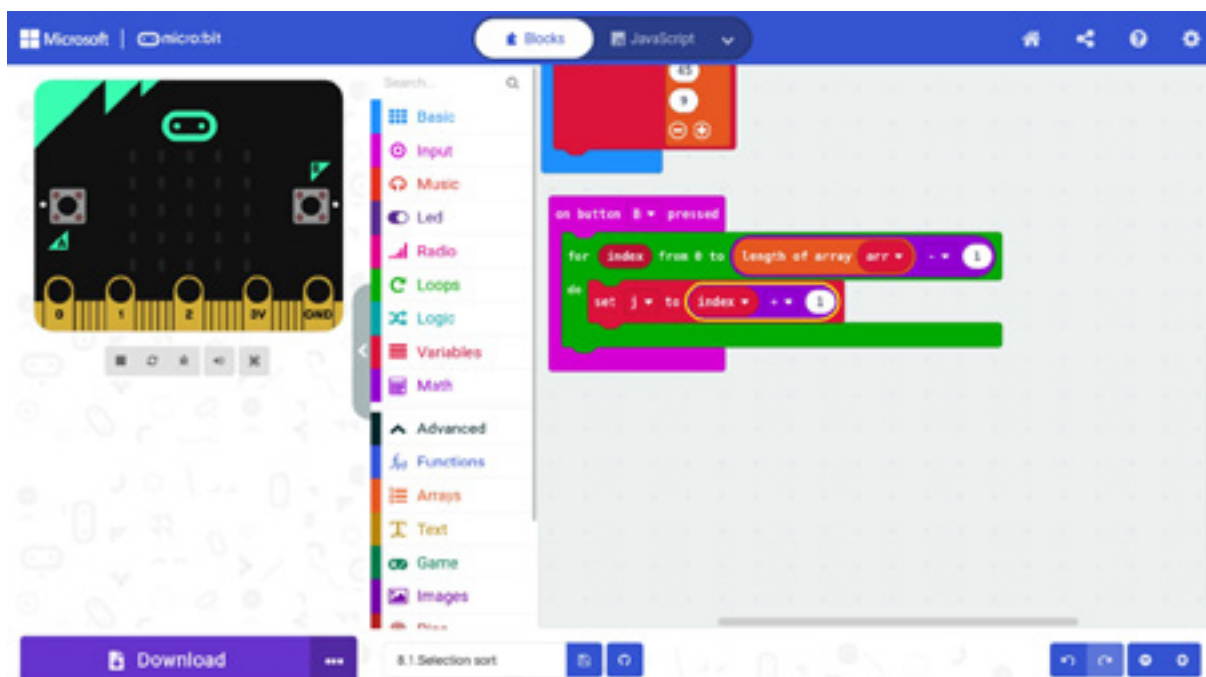
14. Dodaj blok „for index from 0 to ...” u blok „on button B pressed”.



15. Podesi ciklus „for” da se izvršava onoliko puta koliko je dužina niza *arr* umanjena za 1.



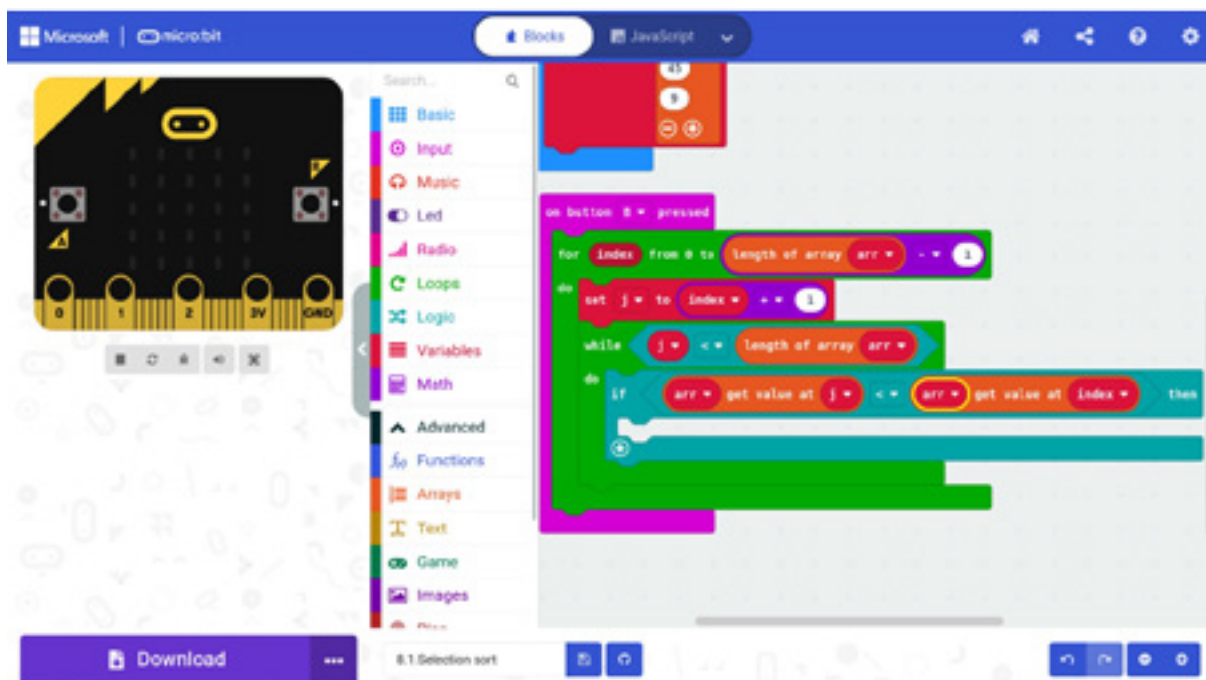
16. Unutar ciklusa *for* postavi vrijednost promjenljive „j” da bude za jedan veća od vrijednosti promjenljive „index”. Promjenljiva „j” će biti indeks elemenata niza *arr*, koji se nalaze iza pozicije zadate promjenljivom „index”.



17. Dodaj ciklus „while” pod uslovom da je vrijednost promjenljive „j” manja od dužine niza *arr*.

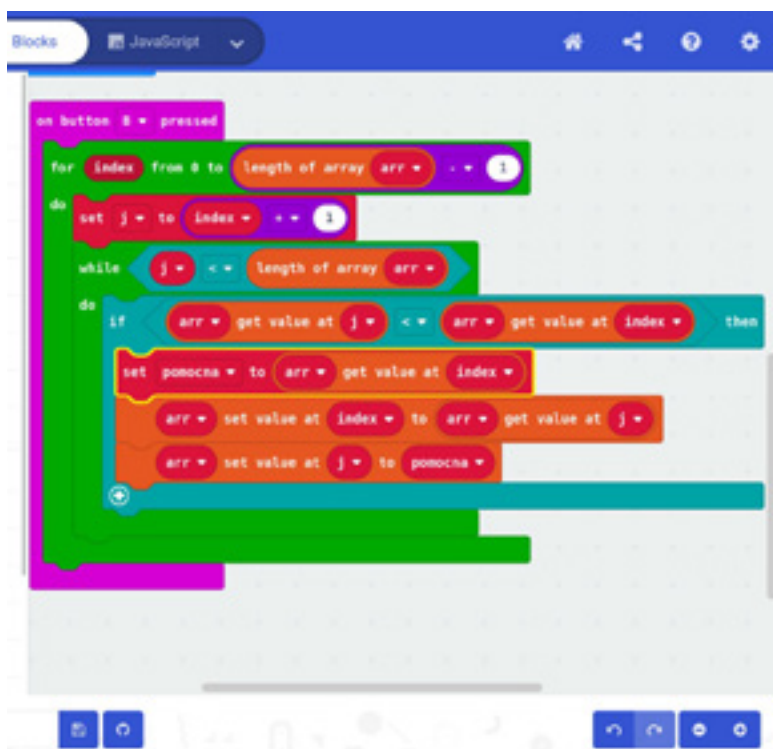


18. Dodaj blok *if* koji provjerava da li je element niza *arr* sa indeksom *j* manji od elementa niza sa indeksom *index*. Koristi blok *arr get value at ...*

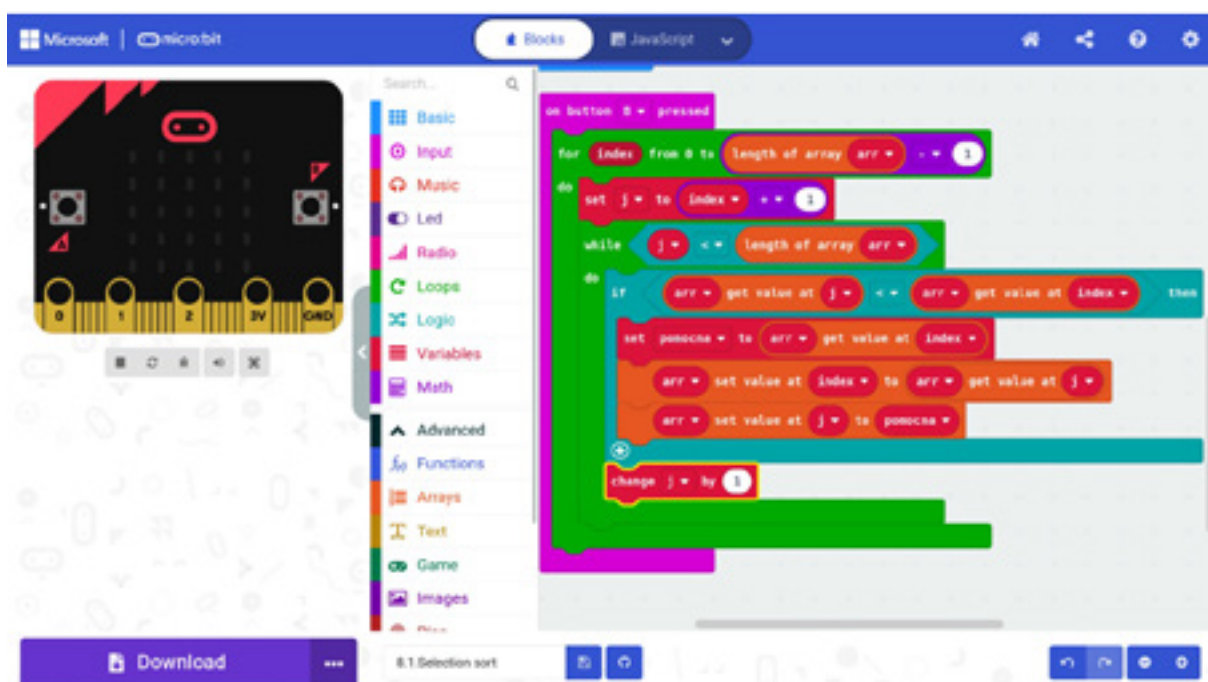


19. Ako je uslov iz naredbe *if* tačan, zamijeni mjesta elementima niza na pozicijama *index* i *j*, koristeći promjenljivu „pomocna”. Potrebna su ti tri bloka, koja predstavljaju sljedeće operacije:

```
pomocna = arr[index]
arr[index] = arr[j]
arr[j] = pomocna
```



20. Poslije naredbe *if*, dodaj blok kojim se vrijednost promjenljive „j” uvećava za 1.



21. Testiraj program. Pritiskom na dugme B, elementi niza će biti sortirani od najmanjeg ka najvećem. Pritiskom na dugme A, na ekranu će se iscrtati elementi niza, od najmanjeg ka najvećem. Sada će niz *arr* imati sljedeći izgled: [6, 8, 9, 12, 22, 23, 34, 45].

22. Prebaci program sa tvog računara na Micro:bit uređaj.

Dodatni materijali:

<https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/a/selection-sort-pseudocode>

https://en.wikipedia.org/wiki/Selection_sort

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_gU26eFKuiJE8



PLAN ČASA BROJ 4.1.	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit uređaj.
KRATAK OPIS ČASA	
<p>Učenici se upoznaju sa načinom sortiranja niza. Naučiće jedan jednostavan metod sortiranja, koji se naziva sortiranje selekcijom (engl. <i>selection sort</i>). Metod se zasniva na podjeli niza na sortirani i nesortirani dio, na pronalaženju najmanjeg elementa u nesortiranom dijelu niza i na zamjeni mjesta elemenata niza. Algoritam za ovaj metod sortiranja je sljedeći:</p> <ol style="list-style-type: none"> 1. Odredi najmanji elemenat u nizu. Zamijeni mjesta prvom elementu niza, tj. onom sa indeksom 0, i najmanjem elementu niza. 2. Odredi drugi najmanji elemenat u nizu. Zamijeni mjesta drugom elementu niza i drugom najmanjem elementu niza. 3. Odredi treći najmanji elemenat u nizu. Zamijeni mjesta trećem elementu niza i trećem najmanjem elementu niza. 4. Ponavlja postupak određivanja sljedećeg najmanjeg elementa i zamjene mjesta, sve dok niz ne bude uređen. 	
TOK ČASA PREMA AKTIVNOSTIMA UČENIKA	

Aktivnost 1

Demonstracija algoritma sortiranja – Osmam učenika dobija po jedan list papira. Na svakom listu papira je zapisan po jedan element niza [12, 8, 23, 6, 34, 22, 45, 9]. Učenici stoje u vrsti i prate instrukcije nastavnika. Deveti učenik, označimo ga sa AAA, stoji ispred prvog učenika, leđima prema učionici i pokazuje prstom na prvog učenika. Ostali učenici traže koji od učenika iz vrste, koji se nalaze desno od AAA, drži najmanji broj, tj. na kojoj se poziciji u vrsti nalazi najmanji broj, i izgovaraju glasno njegov indeks, koji označimo sa IND. Tada učenik na koga pokazuje AAA i učenik IND mijenjaju mjesta. Zatim se AAA pomjeri tako da pokazuje na sljedećeg učenika u vrsti.

Ostali učenici ponavljaju postupak: traže koji od učenika iz vrste, koji se nalaze desno od AAA, drži najmanji broj, tj. na kojoj se poziciji u vrsti nalazi najmanji broj, i izgovaraju glasno njegov indeks, koji označimo sa IND. Tada učenik na koga pokazuje AAA i učenik IND mijenjaju mjesta. Ovaj postupak se ponavlja sve dok učenik AAA ne dođe do posljednjeg učenika u vrsti. Učenici zaključuju da učenici u vrsti sada drže papire koji prikazuju rastući niz brojeva.

Umjesto ove aktivnosti, učenici mogu u svesci zapisati jedan od sljedeća dva primjera, prateći instrukcije nastavnika.

Primjer: Žuta boja označava poziciju na kojoj treba da bude sljedeći najmanji element.

Niz	Najmanji element	Niz poslije zamjene
[12, 8, 23, 6, 34, 22, 45, 9]	6	[6, 8, 23, 12, 34, 22, 45, 9]
[6, 8, 23, 12, 34, 22, 45, 9]	8	[6, 8, 23, 12, 34, 22, 45, 9]
[6, 8, 23, 12, 34, 22, 45, 9]	9	[6, 8, 9, 12, 34, 22, 45, 23]
[6, 8, 9, 12, 34, 22, 45, 23]	12	[6, 8, 9, 12, 34, 22, 45, 23]
[6, 8, 9, 12, 34, 22, 45, 23]	22	[6, 8, 9, 12, 22, 34, 45, 23]
[6, 8, 9, 12, 22, 34, 45, 23]	23	[6, 8, 9, 12, 22, 23, 45, 34]
[6, 8, 9, 12, 22, 23, 45, 34]	34	[6, 8, 9, 12, 22, 23, 34, 45]
[6, 8, 9, 12, 22, 23, 34, 45]	45	[6, 8, 9, 12, 22, 23, 34, 45]
[6, 8, 9, 12, 22, 23, 34, 45]		

Primjer 2:

Sortirani dio niza	Nesortirani dio niza	Najmanji element u nesortiranom dijelu
[]	[12, 8, 23, 6, 34, 22, 45, 9]	6
[6]	[12, 8, 23, 34, 22, 45, 9]	8
[6, 8]	[12, 23, 34, 22, 45, 9]	9
[6, 8, 9]	[12, 23, 34, 22, 45]	12
[6, 8, 9, 12]	[23, 34, 22, 45]	22
[6, 8, 9, 12, 22]	[23, 34, 45]	23
[6, 8, 9, 12, 22, 23]	[34, 45]	34
[6, 8, 9, 12, 22, 23, 34]	[45]	45
[6, 8, 9, 12, 22, 23, 34, 45]	[]	

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa jednostavnom metodom sortiranja i objašnjava učenicima zahtjeve zadatka koje će raditi u daljem dijelu časa. Potrebno je napraviti kôd koji će pritiskom na taster A prikazati niz *arr*, dok će se pritiskom na dugme B prikazati sortirani rastući niz *arr*. Učenici zapisuju niz slučajno odabranih brojeva 12, 8, 23, 6, 34, 22, 45, 9, čije je sortiranje potrebno izvršiti.

Aktivnost 3

Otvaranje sekcije za nizove (Advanced/Arrays) – Učenici, prateći instrukcije nastavnika,

u okviru programa MakeCode otvaraju sekciju za nizove. Klikom na *Advanced*, otvara se lista dodatnih potkategorija i bira se *Arrays*. Biraju blok *set list to array...*



Aktivnost 4

Promjena imena promjenljivih – Na radnoj površini u blok *on start* dodaju blok *set list to array...* i mijenjaju ime niza (promjenljive) u *arr*.

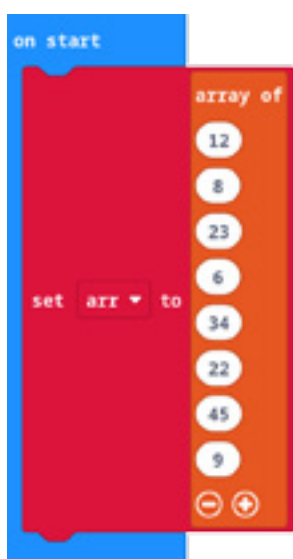


Aktivnost 5

Dodavanje novih elemenata niza – Učenici u okviru bloka *set list to array...*, klikom na +, dodaju nove elemente niza.



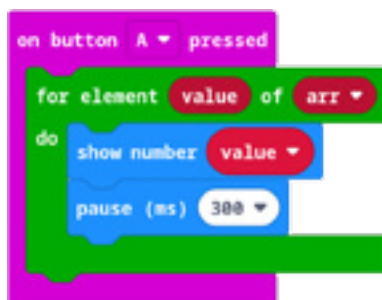
Niz *arr* sadrži 8 elemenata, koje učenici zapisuju na sljedeći način: [12, 8, 23, 6, 34, 22, 45, 9]. Nastavnik naglašava da će indeksi niza *arr* imati vrijednosti 0, 1, 2, 3, 4, 5, 6 i 7. Na primjer, pišemo $arr[1] = 8$ ili $arr[6] = 45$.



Aktivnost 6

Prikaz niza elemenata *arr* na ekranu Micro:bit-a – Učenici kreiraju kôd koji pritiskom na dugme A prikazuje elemente niza *arr*, dok će se pritiskom na dugme B sortirati niz *arr* u rastući.

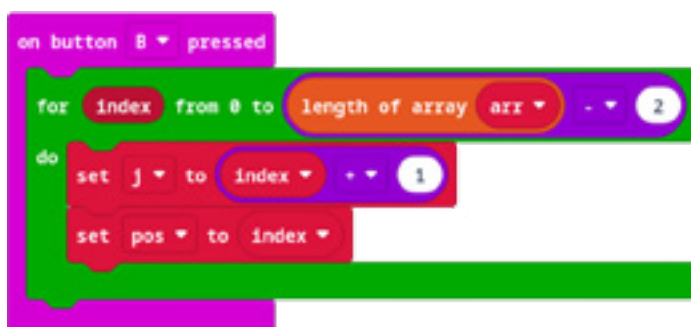
U blok *on button A pressed* dodaju blok *for element value of arr* i biraju niz *arr*. U ciklus *for* dodaju blok za prikazivanje elementa niza (tj. *value*) i zatim pauzu od 300 ms.



Aktivnost 7

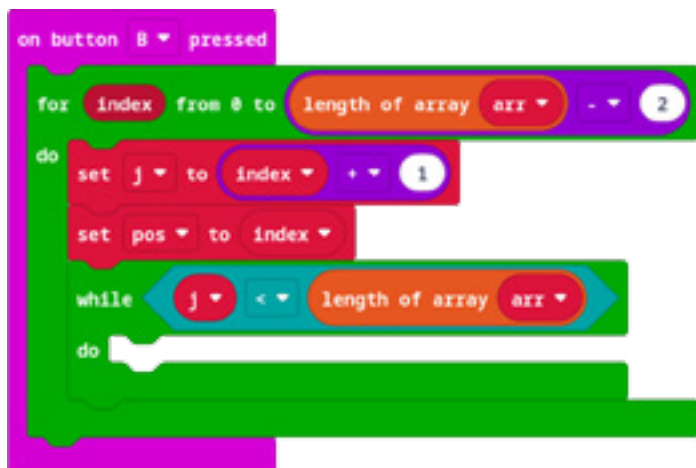
Sortiranje niza – Za potrebe sortiranja niza potrebno je da učenici definišu tri nove promjenljive: „*j*“, koja će imati ulogu indeksa niza, „*pomocna*“, čija će uloga biti da im pomogne prilikom zamjene mjesta elemenata niza i „*pos*“, koja sadrži poziciju najmanjeg elementa niza u nesortiranom dijelu niza. Sortiranje se vrši klikom na taster B, te kôd sortiranja učenici smještaju unutar bloka *on button B pressed*.

Dodaju blok *for index from 0 to ...* i podešavaju ciklus *for* da se izvršava onoliko puta kolika je dužina niza *arr* umanjena za 2. Unutar ciklusa *for* postavljaju vrijednost promjenljive „*j*“ da bude za jedan veća od vrijednosti promjenljive „*index*“. Promjenljiva „*j*“ će biti indeks elemenata niza *arr*, koji se nalaze iza pozicije zadate promjenljivom „*index*“. Promjenljiva „*pos*“ dobija vrijednost promjenljive „*index*“.



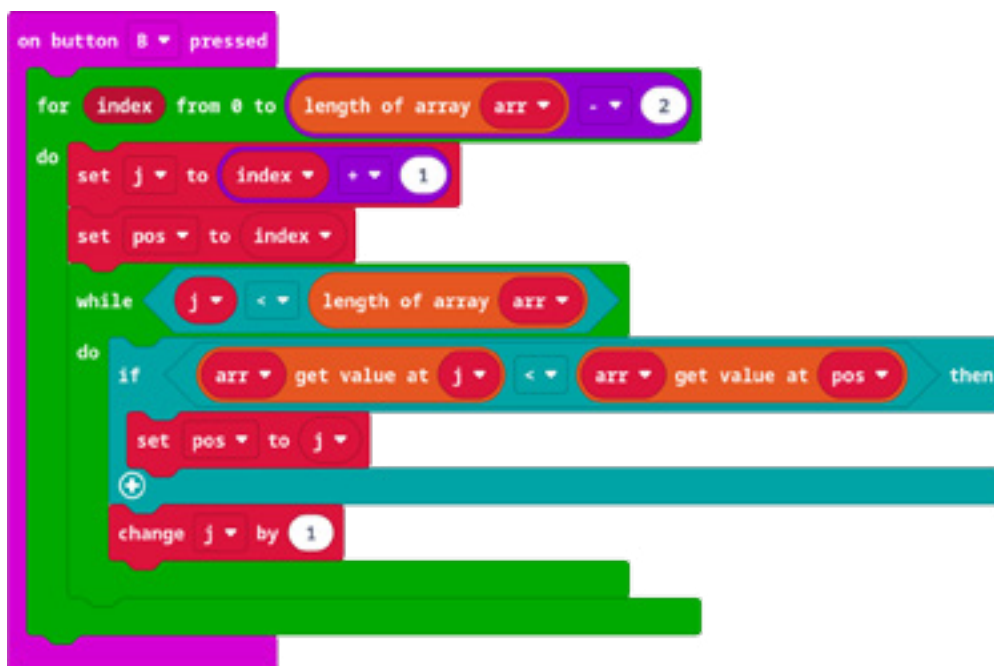
Aktivnost 8

Sortiranje niza – Učenici dodaju ciklus *while* pod uslovom da je vrijednost promjenljive „*j*“ manja od dužine niza *arr*.



Aktivnost 9

Upoređivanje elemenata niza – Učenici dodaju blok *if* koji provjerava da li je element niza *arr* sa indeksom *j* manji od elementa niza sa indeksom *pos*. Koriste blok *arr get value at ...*. U bloku *if* postavljaju vrijednost promjenljive „pos“ na „j“, a poslije bloka *if* uvećavaju *j* za 1.



Aktivnost 10

Zamjena mjesta elemenata – Poslije bloka *while*, učenici mijenjaju mjesta elementima niza na pozicijama *pos* i *index*, koristeći promjenljivu „pomocna“. Učenici koriste tri bloka, koja predstavljaju sljedeće operacije:

```
pomocna = arr[index]
arr[index] = arr[pos]
arr[pos] = pomocna
```



Aktivnost 11

Učenici testiraju program na simulatoru, uklanjaju moguće greške i prenose ispravnu verziju na Micro:bit uređaj.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_8AxUoUc4LR0v



PLAN ČASA BROJ 4.1.1.	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit uređaj.
KRATAK OPIS ČASA	
Učenici se upoznaju sa načinom sortiranja niza. Naučiće jedan jednostavan metod sortiranja, koji se naziva sortiranje selekcijom (engl. <i>selection sort</i>). Metod se zasniva na podjeli niza na sortirani i nesortirani dio, na pronalaženju najmanjeg elementa u nesortiranom dijelu niza i zamjeni mjesta elemenata niza.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Demonstracija algoritma sortiranja – Osam učenika dobija po jedan list papira. Na svakom listu papira je zapisan po jedan element niza [12, 8, 23, 6, 34, 22, 45, 9]. Učenici stoje u vrsti i prate instrukcije nastavnika. Deveti učenik, označimo ga sa AAA, stoji ispred prvog učenika, leđima prema učionici i pokazuje prstom na prvog učenika. Ostali učenici posmatraju redom sve učenike desno od učenika na kojeg pokazuje AAA. Svaki put kada naiđu na učenika koji drži broj koji je manji od broja koji drži učenik na koga pokazuje AAA, traže da ta dva učenika zamijene mjesta. Ponavljaju postupak dok ne dođu do kraja vrste. Zatim se AAA pomjeri tako da pokazuje na sljedećeg učenika u vrsti.

Ostali učenici ponavljaju opisani postupak: posmatraju učenike iz vrste koji se nalaze desno od AAA, i traže da učenici u vrste zamjenjuju mjesta ako brojevi koje drže nijesu u odgovarajućem položaju. Ovaj postupak se ponavlja sve dok učenik AAA ne dođe do posljednjeg učenika u vrsti. Učenici zaključuju da učenici u vrsti sada drže papire koji prikazuju rastući niz brojeva.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima otvaraju radni list „RadniListSelectionSort.pdf” sa lokacije na koju ih uputi nastavnik.

Aktivnost 3

Učenici čitaju sekcije Uvod i Primjer 1 radnog lista.

Aktivnost 4

Pokreću program MakeCode i kreiraju novi projekat sa nazivom „8.1 Selection Sort“.

Aktivnost 5

Implementiraju korake 1–10 radnog lista. Obavještavaju nastavnika da su završili sve korake, pa zajedno provjeravaju ispravnost programa izvršavanjem koraka 11 radnog lista.

Aktivnost 6

Implementiraju korake 12–16 radnog lista. Obavještavaju nastavnika da su završili sve korake, pa zajedno provjeravaju ispravnost programa.

Aktivnost 7

Implementiraju korake 17–21 radnog lista. Obavještavaju nastavnika da su završili sve korake i samostalno provjeravaju ispravnost programa.

Aktivnost 8

Ako u koraku 21 ne dobiju željeni rezultat, samostalno pokušavaju da uoče moguće greške i uklone ih.

Aktivnost 9

Učenici kojima je program ispravan izvršavaju korak 22 radnog lista. Ostali učenici, uz pomoć nastavnika, pokušavaju da lociraju grešku u programu.

PLAN ČASA BROJ 4.2.	
OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode.
KRATAK OPIS ČASA	
Učenici se upoznaju sa načinom sortiranja niza. Naučiće jedan jednostavan metod sortiranja, koji se naziva BubbleSort. Upoređujemo susjedne elemente niza i ako nijesu u odgovarajućem poretku, zamjenjujemo im mjesta. Ovaj postupak ponavljamo sve dok ima zamjene mjesta u nizu. Sortiranje se obično obavlja u samom nizu (engl. <i>in-place</i>), bez uvođenja pomoćnog niza.	

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Demonstracija algoritma sortiranja – Pet učenika dobija po jedan list papira. Na svakom listu papira je zapisan po jedan element niza [5, 1, 4, 2, 8]. Učenici prate instrukcije nastavnika. Sve dok niz nije sortiran, ponavljaju sljedeće radnje: svaka dva susjedna učenika koja nijesu u pravom poretku mijenjaju mjesta, kao u primjeru 1. Odgovaraju na pitanje: „Koliko je ukupno zamjena potrebno u najmanje povoljnom slučaju?”

Umjesto ove aktivnosti, učenici mogu zapisati u sveskama sljedeći algoritam, dat pseudokodom i uraditi primjer. Umjesto pseudokoda, mogu nacrtati dijagram toka.

A je niz koji ima n elemenata sa indeksima 0, ..., n-1:

```
function Bubble_sort A
  n ← length(A)
  repeat
    swapped ← false /* da li se dogodila zamjena mjesta elemenata */
    for i ← 1 to n-1 do
      /* ako elementi nijesu u pravom poretku */
      if A[i-1] > A[i] then
        /* zamijeni mjesta A[i-1] i A[i] */
        t ← A[i-1]
        A[i-1] ← A[i]
        A[i] ← t
        swapped ← true
      end if
    end for
  until not swapped
```

Primjer 1: Prikazaćemo izvršavanje algoritma *Bubble Sort* na primjeru niza [5, 1, 4, 2, 8].

Prvi prolaz:

[5, 1, 4, 2, 8] → [1, 5, 4, 2, 8] – zamjena prva dva elementa jer je $5 > 1$

[1, 5, 4, 2, 8] → [1, 4, 5, 2, 8] – zamjena drugog i trećeg elementa jer je $5 > 4$

[1, 4, 5, 2, 8] → [1, 4, 2, 5, 8] – zamjena jer je $5 > 2$

[1, 4, 2, 5, 8] → [1, 4, 2, 5, 8] – nema zamjene jer je $8 > 5$

Drugi prolaz:

[1, 4, 2, 5, 8] → [1, 4, 2, 5, 8]

[1, 4, 2, 5, 8] → [1, 2, 4, 5, 8] – zamjena jer je $4 > 2$

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

Treći prolaz:

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

[1, 2, 4, 5, 8] → [1, 2, 4, 5, 8]

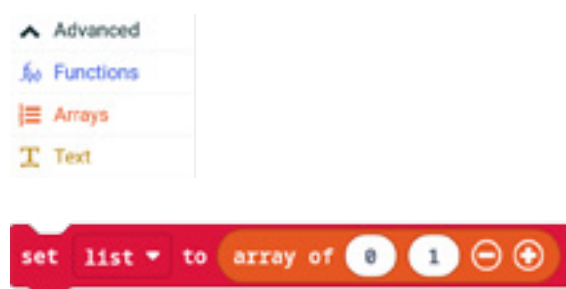
Nije bilo zamjene mjesta, pa algoritam završava rad.

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa jednostavnom metodom sortiranja i objašnjava učenicima zahtjeve zadatka koje će raditi u nastavku. Potrebno je napraviti kôd koji će pritiskom na taster A prikazati niz *list*, dok će se pritiskom na dugme B sortirati niz *list* u rastući. Saopštava niz slučajno odabranih brojeva 12, 8, 23, 6, 34, 22, 45, 9, čije je sortiranje potrebno izvršiti.

Aktivnost 3

Otvaranje sekcije za nizove (Advanced/Arrays) – Učenici, prateći instrukcije nastavnika, u okviru programa MakeCode otvaraju sekciju za nizove. Klikom na *Advanced*, otvara se lista dodatnih potkategorija i bira se *Arrays*. Učenici biraju blok *set list to array...*

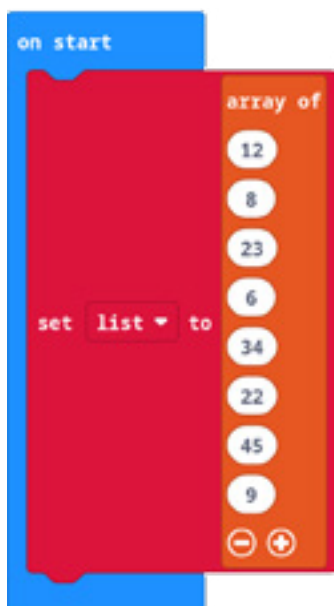


Aktivnost 4

Dodavanje novih elemenata niza – Učenici u okviru bloka *set list to array...*, klikom na +, dodaju nove elemente niza.



Niz *list* sadrži 8 elemenata, koje učenici zapisuju na sljedeći način: [12, 8, 23, 6, 34, 22, 45, 9]. Nastavnik podsjeća učenike da će indeksi niza *list* imati vrijednosti 0, 1, 2, 3, 4, 5, 6 i 7. Na primjer, pišemo `list[1] = 8` ili `list[6] = 45`.



Aktivnost 5

Opis zadatka – Učenici saznaju da treba napisati program koji pritiskom na dugme A poziva funkciju za štampanje (prikaz) elemenata niza *list*, dok pritiskom na dugme B poziva funkciju koja će sortirati niz *list* u rastući. Učenici odgovaraju na pitanje: „Kako se kreira funkcija u programu MakeCode?“ Podsjećaju se pojmovu parametar (argument) funkcije i uočavaju da niz može biti argument funkcije. Uočavaju da se prikaz elemenata niza može implementirati primjenom funkcije.

Aktivnost 6

Prikaz niza elemenata „list“ na ekranu Micro:bit-a – Učenici biraju sekciju *Functions*, kreiraju funkciju *stampaNiza* pomoću *Make function* i postavljaju da funkcija ima jedan parametar tipa *Array*, čije je ime *lst*. Mogu zadati i drugo ime funkcije i argumenta, ali pri implementaciji funkcije neophodno je navesti odgovarajuća imena.

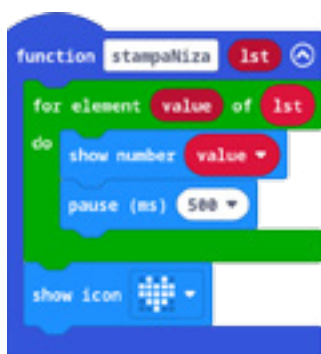
U blok *on button A pressed* dodaju blok *call function stampaNiza* i biraju niz *list* za parametar funkcije.

Ovaj dio učenici mogu uraditi i bez primjene funkcije, ali preporučujemo da se zadatak rješava primjenom funkcije.



Aktivnost 7

Implementacija funkcije *stampaNiza* – Ako su učenici već pisali program za Selection sort, mogu kopirati blok iz drugog projekta u funkciju *stampaNiza*. U suprotnom, dodati ciklus *for element value of lst* i unutar njega blok za prikazivanje broja *show number* i pauzu od 500 milisekundi. Obratiti pažnju da parametar funkcije bude u ciklusu *for* (tj. parametar je *lst*). Iza ciklusa dodati blok *show icon*, koji prikazuje simbol srca, čime je označen kraj izvršavanja funkcije.



Aktivnost 8

Sortiranje niza – Učenici ponavljaju postupak kreiranja funkcije, samo što sada kreiraju funkciju *BubbleSort* pomoću *Make function* i postavljaju da funkcija ima jedan parametar tipa *Array*, čije je ime *lst*. Za potrebe sortiranja niza potrebno je da učenici definišu novu promjenljivu „swapped“, čija vrijednost može biti 1, ako se dogodila bar jedna zamjena mjesta u nizu *lst* ili 0, ako nije bilo zamjene. Učenici dodaju tzv. „beskonačni ciklus“, koji će biti prekinut kada promjenljiva „swapped“ dobije vrijednost 0. „Beskonačni ciklus“ se može realizovati pomoću bloka *while true*; unutar ciklusa dodaje se blok *if swapped = 0*, koji će prekinuti izvršavanje ciklusa pozivom bloka *break*.

Aktivnost 9

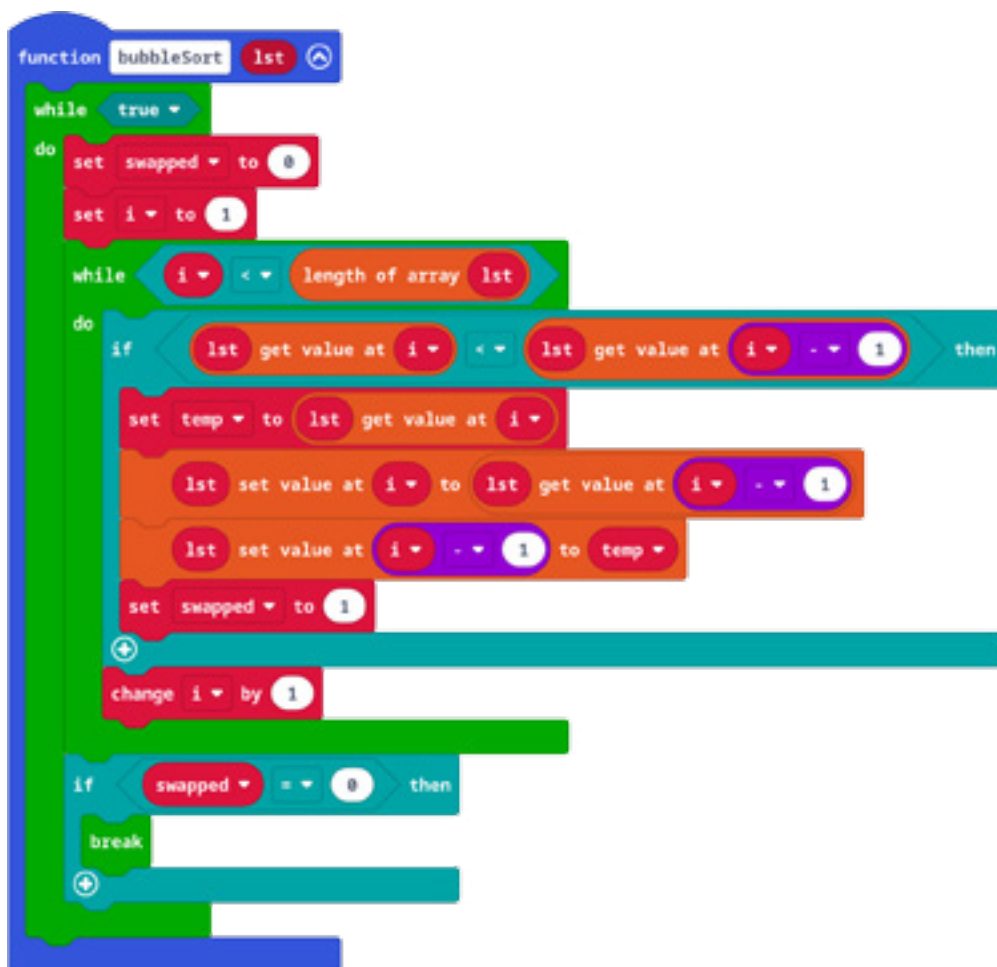
Poređenje elemenata niza – Učenici dodaju ciklus kojim provjeravaju da li su dva uzastopna elementa, tj. oni sa indeksima $i-1$ i i , u pravom poretku. Ako nijesu, zamjenjuju im mjesta koristeći pomoćnu promjenljivu „temp“.

Učenici kreiraju promjenljive „i“ i „temp“. Prvo dodaju blok koji promjenljivoj „i“ dodjeljuje vrijednost 1, pa zatim dodaju ciklus *while* koji provjerava da li je vrijednost promjenljive „i“ manja od dužine niza *lst* (tj. koriste blok *length array of lst*, gdje je *lst* parametar funkcije). Unutar bloka *while* dodaju blok *if*, kojim upoređuju elemente niza *lst* sa indeksima $i-1$ i i , koristeći blokove *lst get value of i* i *lst get value of i-1*.

Ako je uslov u bloku *if* tačan, elementi „*lst[i-1]*“ i „*lst[i]*“ zamjenjuju mjesta pomoću promjenljive „temp“ i mijenjaju vrijednost promjenljive „swapped“ na 1. Učenici koriste četiri bloka, koja predstavljaju sljedeće operacije:

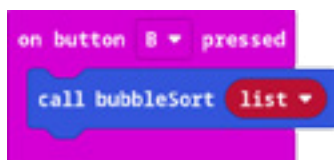
```
temp = lst[index]
lst[index] = lst[j]
lst[j] = temp
swapped = 1
```


Poslije bloka *i* uvećavamo vrijednost promjenljive „i“ za 1 pomoću bloka *change i by 1*.



Aktivnost 10

Pozivanje funkcija – Učenici dodaju blok *on button B pressed* i u njemu pozivaju funkciju *bubbleSort* pomoću bloka *call function BubbleSort*. Učenici mogu pokrenuti simulator i testirati program prvo klikom na dugme A. Mogu se uvjeriti da će biti prikazan polazni niz brojeva. Zatim se pritiskom na dugme B niz sortira. Sljedeći pritisak na dugme A štampa elemente niza u rastućem poretku.



Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_hyXH507wparv



PLAN ČASA BROJ 4.3.

OVI 8	Na kraju učenja učenik će biti osposobljen da piše programe koji komuniciraju u mreži uređaja i koji koriste nizove.
Ishodi učenja	Na kraju učenja učenik će biti osposobljen da kreira programe koji koriste nizove za čuvanje podataka.
Nastavna sredstva i pomagala	Računar, projektor, program MakeCode, Micro:bit uređaj.

KRATAK OPIS ČASA

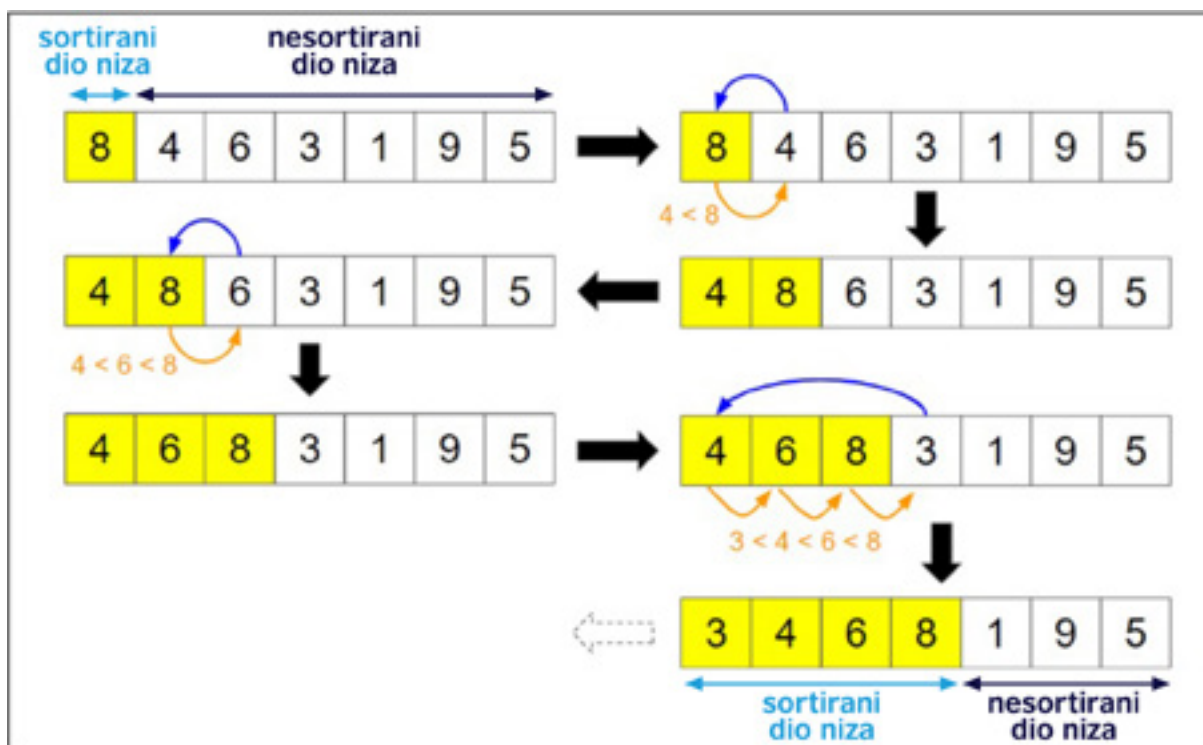
Sortiranje umetanjem (Insertion sort) uklanja elemente jedan po jedan iz nesortiranog dijela niza i umeće ih u sortirani dio niza na odgovarajuću poziciju. Sortiranje se obično obavlja u samom nizu (engl. *in-place*), bez uvođenja pomoćnog niza. Elementi se uklanjaju sa početka ili kraja niza, čime se stvara prostor za sortirani dio niza.

TOK ČASA PREMA AKTIVNOSTIMA UČENIKA

Aktivnost 1

Demonstracija algoritma sortiranja – Osam učenika dobija po jedan list papira. Na svakom listu papira je zapisan po jedan element niza [12, 8, 23, 6, 34, 22, 45, 9]. Učenici stoje u vrsti i prate instrukcije nastavnika. Upoređuju brojeve prvog i drugog učenika i postavljaju ih u rastući poredak. Zatim upoređuju trećeg učenika i postavljaju ga na pravu poziciju, pomjerajući ostale učenike. Učenik koji se umeće u sortirani dio niza istupa iz vrste, da bi napravio mjesto na koje će se pomjeriti učenici ispred njega u vrsti. Ponavljaju postupak sve dok ne dobiju uređeni niz. Umjesto ove aktivnosti, učenici mogu u sveskama da zapišu primjer 1 ili primjer 2, a zatim zapišu algoritam:

Primjer 1: Sortiranje niza sa 7 elemenata primjenom algoritma *Insertion sort*. Označeni su sortirani dio niza (engl. *sorted subarray*) i nesortirani dio niza (engl. *unsorted subarray*). Sortirani dio niza se nalazi na početku niza.



Slika preuzeta sa <https://medium.com/star-gazers/introduction-to-insertion-sort-756821945e3e>.

Primjer 2: Sortiranje niza sa 8 elemenata primjenom algoritma *Insertion sort*. Elementi označeni žutom bojom čine sortirani dio niza, dok bijela boja označava elemente u nesortiranom dijelu niza. Crvena boja označava elemente koji su pomjereni u trenutnom koraku.

Niz	Element koji umećemo	Niz poslije umetanja
[12, 8, 23, 6, 34, 22, 45, 9]	8	[8, 12, 23, 6, 34, 22, 45, 9]
[8, 12, 23, 6, 34, 22, 45, 9]	23	[8, 12, 23, 6, 34, 22, 45, 9]
[8, 12, 23, 6, 34, 22, 45, 9]	6	[6, 8, 12, 23, 34, 22, 45, 9]
[6, 8, 12, 23, 34, 22, 45, 9]	34	[6, 8, 12, 23, 34, 22, 45, 9]
[6, 8, 12, 23, 34, 22, 45, 9]	22	[6, 8, 12, 22, 23, 34, 45, 9]
[6, 8, 12, 22, 23, 34, 45, 9]	45	[6, 8, 12, 22, 23, 34, 45, 9]
[6, 8, 12, 22, 23, 34, 45, 9]	9	[6, 8, 9, 12, 22, 23, 34, 45]

Ako je A niz od n elemenata koji sortiramo, čiji su indeksi 0, ..., n-1, tada se ovaj algoritam može opisati sljedećim pseudokodom:

```
function insertion_sort A
    i ← 1
    while i < n do
        v ← A[i]
        h = i
        /* odredimo poziciju gdje umecemo   Član v */
        while (h > 0 and A[h-1] > v)
            A[h] ← A[h-1]
            h ← h -1
        /* umecemo v na poziciju h */
        A[h] ← v;
        i ← i + 1
```

Aktivnost 2

Pokretanje programa – Učenici na svojim računarima pokreću program MakeCode i prate rad nastavnika. Nastavnik upoznaje učenike sa jednostavnom metodom sortiranja i objašnjava učenicima zahtjeve zadatka koje će raditi u nastavku. Potrebno je napraviti kôd koji će pritiskom na taster A prikazati niz *list*, dok će se pritiskom na dugme B sortirati niz *list* u rastući. Saopštava niz slučajno odabranih brojeva 12, 8, 23, 6, 34, 22, 45, 9, čije je sortiranje potrebno izvršiti.

Aktivnost 3

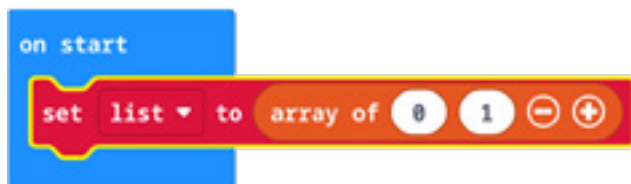
Otvaranje sekcije za nizove (Advanced/Arrays) – Učenici, prateći instrukcije nastavnika, u okviru programa MakeCode otvaraju sekciju za nizove. Klikom na *Advanced*, otvara se lista dodatnih potkategorija i bira se *Arrays*. Biraju blok *set list to array...*



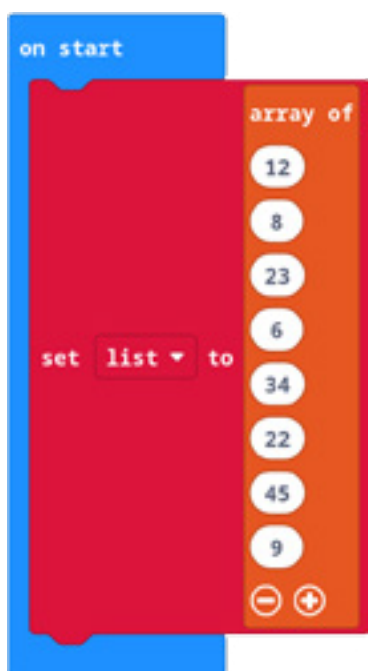


Aktivnost 4

Dodavanje novih elemenata niza – Učenici u okviru bloka *set list to array...*, klikom na +, dodaju nove elemente niza.



Niz *list* sadrži 8 elemenata, koje učenici zapisuju na sljedeći način: [12, 8, 23, 6, 34, 22, 45, 9]. Nastavnik podsjeća učenike da će indeksi niza *list* imati vrijednosti 0, 1, 2, 3, 4, 5, 6 i 7. Na primjer, pišemo `list[1] = 8` ili `list[6] = 45`.



Aktivnost 5

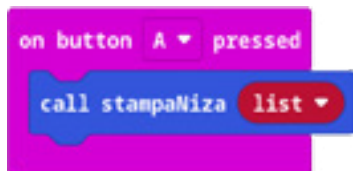
Opis zadatka – Učenici saznaju da treba napisati program koji pritiskom na dugme A poziva funkciju za štampanje (prikaz) elemenata niza *list*, dok pritiskom na dugme B poziva funkciju koja će sortirati niz *list* u rastući. Učenici odgovaraju na pitanje: „Kako se kreira funkcija u programu MakeCode?“. Podsjećaju se pojmova parametar (argument) funkcije i uočavaju da niz može biti argument funkcije. Uočavaju da se prikaz elemenata niza može implementirati primjenom funkcije.

Aktivnost 6

Prikaz niza elemenata „list“ na ekranu Micro:bit-a – Učenici biraju sekciju *Functions*, kreiraju funkciju *stampaNiza* pomoću *Make function* i postavljaju da funkcija ima jedan parametar tipa *Array*, čije je ime *lst*. Mogu zadati i drugo ime funkcije i argumenta, ali pri implementaciji funkcije neophodno je navesti odgovarajuća imena.

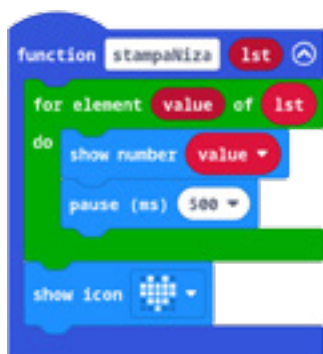
U blok *on button A pressed* dodaju blok „*call function stampaNiza*” i biraju niz „*list*” za parametar funkcije.

Ovaj dio učenici mogu uraditi i bez primjene funkcije, ali preporučujemo da se zadatak rješava primjenom funkcije.



Aktivnost 7

Implementacija funkcije *stampaNiza* – Ako su učenici već pisali program za Selection sort ili BubbleSort, mogu kopirati blok iz drugog projekta u funkciju *stampaNiza*. U suprotnom, dodati ciklus *for element value of lst* i unutar njega blok za prikazivanje broja *show number* i pauzu od 500 milisekundi. Obratiti pažnju da parametar funkcije bude u ciklusu *for* (tj. parametar je „*lst*”). Iza ciklusa dodati blok *show icon*, koji prikazuje simbol srca, čime je označen kraj izvršavanja funkcije.



Aktivnost 8

Sortiranje niza – Učenici ponavljaju postupak kreiranja funkcije, samo što sada kreiraju funkciju *insertionSort* pomoću *Make function* i postavljaju da funkcija ima jedan parametar tipa *Array*, čije je ime *lst*. Za potrebe sortiranja niza potrebno je da učenici definišu nove promjenljive „*i*” i „*temp*”. Promjenljiva „*i*” će, uz promjenljivu „*index*”, služiti za indekse niza „*lst*”, dok će se u „*temp*” čuvati vrijednost elementa koji se umeće u sortirani dio niza. Učenici dodaju blok koji postavlja vrijednost promjenljive „*i*” na 1.

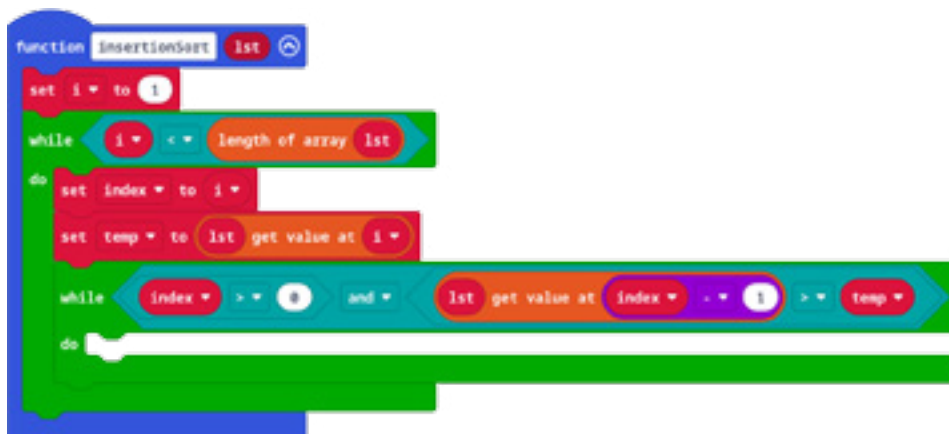
Zatim dodaju blok ciklusa *while* u kojem provjeravaju da li promjenljiva „*i*” i dalje predstavlja indeks niza, tj. da li je i manje od dužine niza..



Aktivnost 9

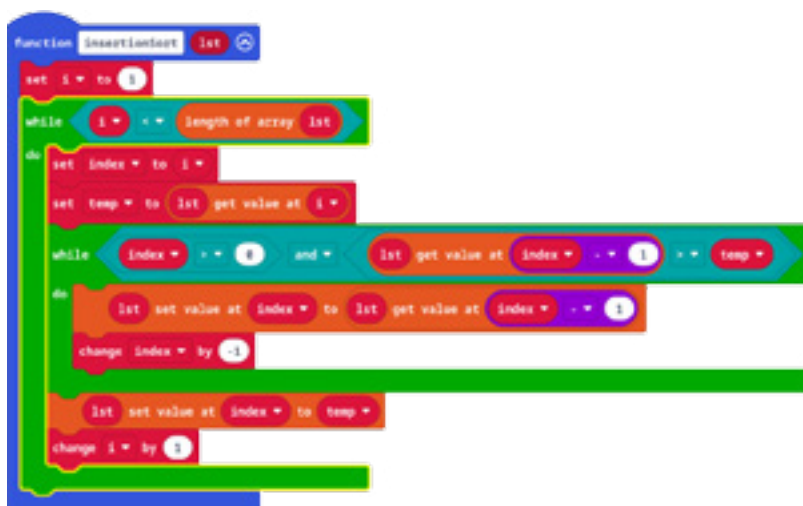
Poređenje elemenata niza – Učenici dodaju blokove koji postavljaju vrijednost promjenljive „index” na „i” i vrijednost promjenljive „temp” na „lst”[i]. Zatim dodaju blok za ciklus *while*, kojim provjeravaju da li važe sljedeća dva uslova:

1. promjenljiva „index” nije došla do prvog indeksa niza, tj. $\text{index} > 0$;
2. element sa indeksom index-1 je veći od „temp”, tj. $(\text{lst get value at } (\text{index}-1)) > \text{temp}$.



Aktivnost 10

Pomjeranje elemenata niza – Učenici u unutrašnji ciklus dodaju blok koji pomjera elemente tj. „lst”[i] = „lst”[i-1] i blok koji mijenja vrijednost promjenljive „index” za 1. Po završetku unutrašnjeg ciklusa, u spoljašnjem ciklusu, dodaju blok koji u niz *lst* na poziciju „index” postavlja vrijednost promjenljive „temp”, a zatim i blok koji povećava promjenljivu „i” za 1.



Aktivnost 11

Pozivanje funkcija – Učenici dodaju blok *on button B pressed* i u njemu pozivaju funkciju *insertionSort* sa argumentom „lst” pomoću bloka *call function insertionSort*. Učenici mogu pokrenuti simulator i testirati program prvo klikom na dugme A. Mogu se uvjeriti da će biti prikazan polazni niz brojeva. Zatim se pritiskom na dugme B niz sortira. Sljedeći pritisak na dugme A štampa elemente niza u rastućem poretku. Poslije provjere na simulatoru, učenici prenose program sa računara na Micro:bit uređaj.

Program možete pogledati kada skenirate QR kôd ili na linku:
https://makecode.microbit.org/_TirDj1UrLdvM



DODATAK 2 – PRILOG ZA PROJEKAT „SIMULACIJA SLUČAJNIH DOGAĐAJA”

Simulacija slučajnih događaja

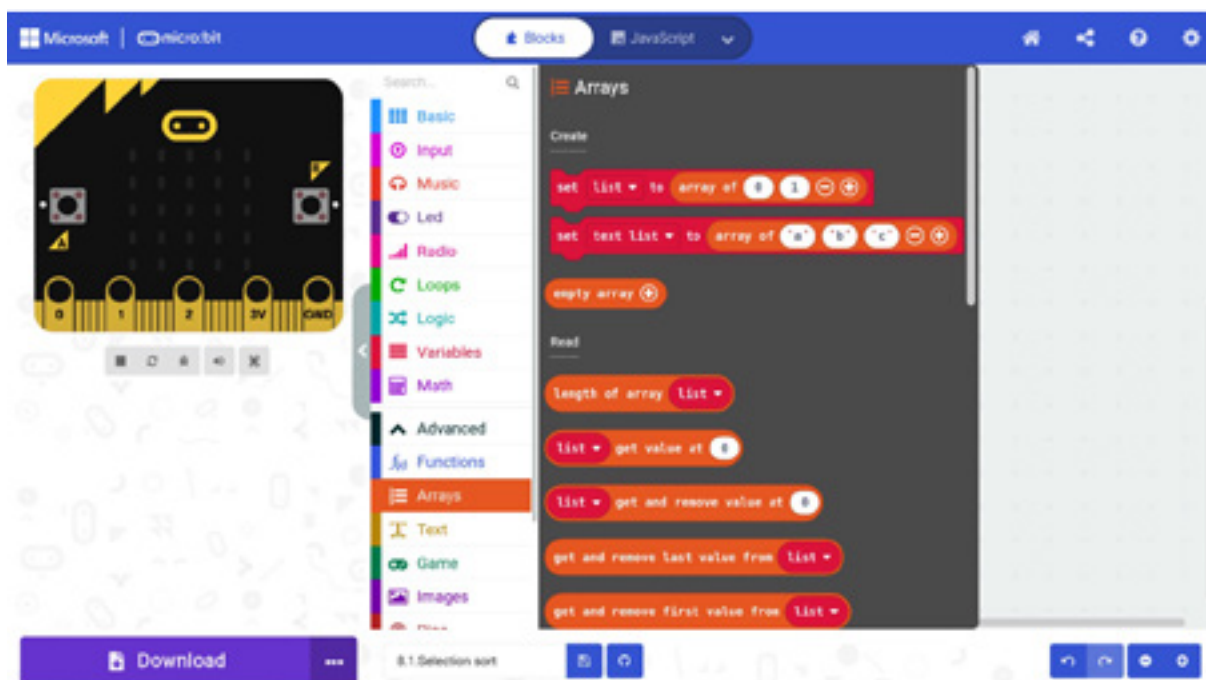
Uvod

Prilikom bacanja novčića, moguće je da dobijemo glavu (engl. *head*) ili pismo (engl. *tail*). Slično, prilikom bacanja kockice za igru „Ne ljuti se, čovječe” (ili Monopol ili Jamb...), možemo dobiti jedan od brojeva 1, 2, 3, 4, 5 ili 6, sa jednakom šansom. Realan broj koji nam kaže kolika je šansa da dobijemo, na primjer, broj 4 pri bacanju jedne kockice naziva se vjerovatnoćom događaja. Tako je vjerovatnoća događaja da dobijemo jednu peticu bacanjem kockice jednaka $1/6$ (od 6 mogućih ishoda, samo jedan nam odgovara). Slično, vjerovatnoća $1/6$ se dobija za bilo koji od brojeva 1, 2, 3, 4, 5 i 6. Vjerovatnoća da dobijemo pismo bacanjem novčića je 0,5, tj. $1/2$.

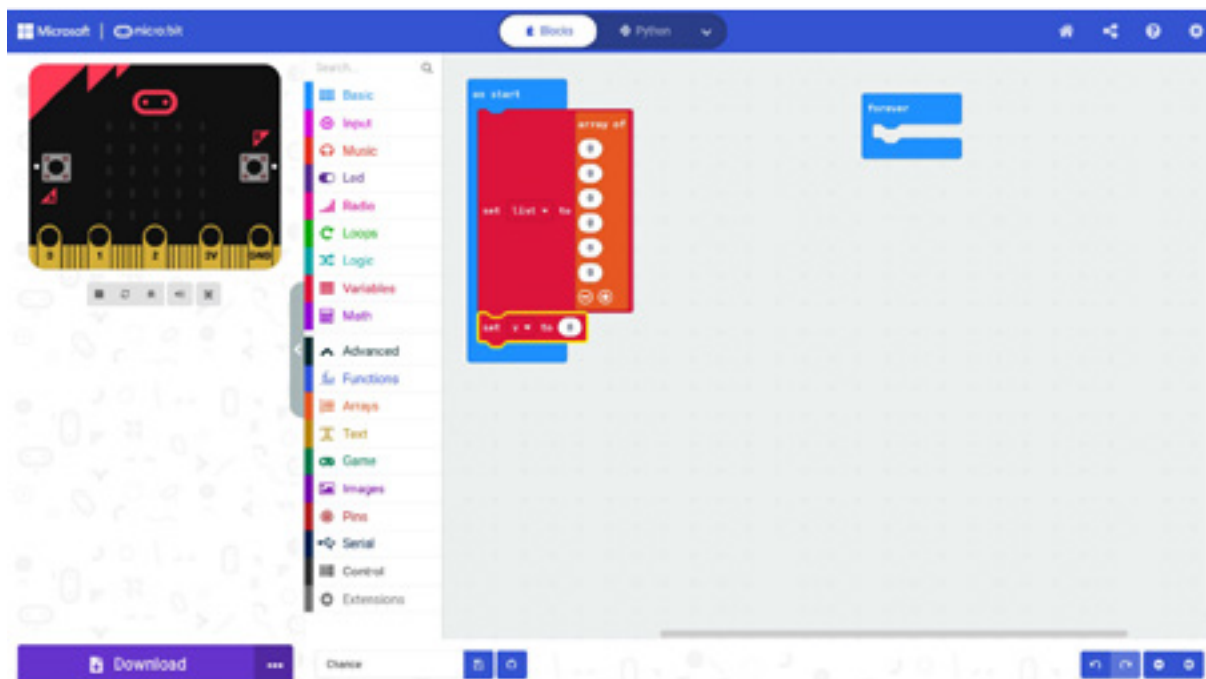
Možemo napraviti i složenije događaje i odrediti njihove vjerovatnoće. Na primjer, vjerovatnoća da istovremenim bacanjem dvije kockice dobijemo dvije šestice iznosi $1/36$, tj. $1/6 \cdot 1/6$. Slično, vjerovatnoća da je zbir brojeva na te dvije kockice jednak 11 iznosi $2/36$. Slučajni ili stohastički proces je matematički opis slučajnih događaja koji se dešavaju jedan za drugim, pa ih je moguće sortirati po vremenima kada su se dogodili. Koriste se za kreiranje modela kretanja cijena na berzi ili za medicinske informacije kao što su EKG, EEG, krvni pritisak ili temperatura.

Primjenom uređaja Micro:bit, simuliraćemo eksperiment bacanja kockica određeni broj puta i na osnovu rezultata procijeniti vjerovatnoću za svaki mogući ishod. Procjena vjerovatnoće se dobija kao količnik broja pojavljivanja nekog ishoda i broja ponavljanja eksperimenta. Na primjer, ako bacanjem kockice 30 puta dobijemo 6 puta peticu, tada naša procjena vjerovatnoće da je pri jednom bacanju „pala” petica iznosi $6/30$, tj. $1/5$, što se razlikuje od vjerovatnoće $1/6$. Ako ponovimo eksperiment i bacamo kockicu 300 puta, možemo dobiti, na primjer, 56 petica, pa će ovaj put naša procjena biti $56/300$, što je još bliže vjerovatnoći $1/6$. Što više puta bacamo kockicu, odnos broja petica i ukupnog broja bacanja biće sve bliži broju $1/6$.

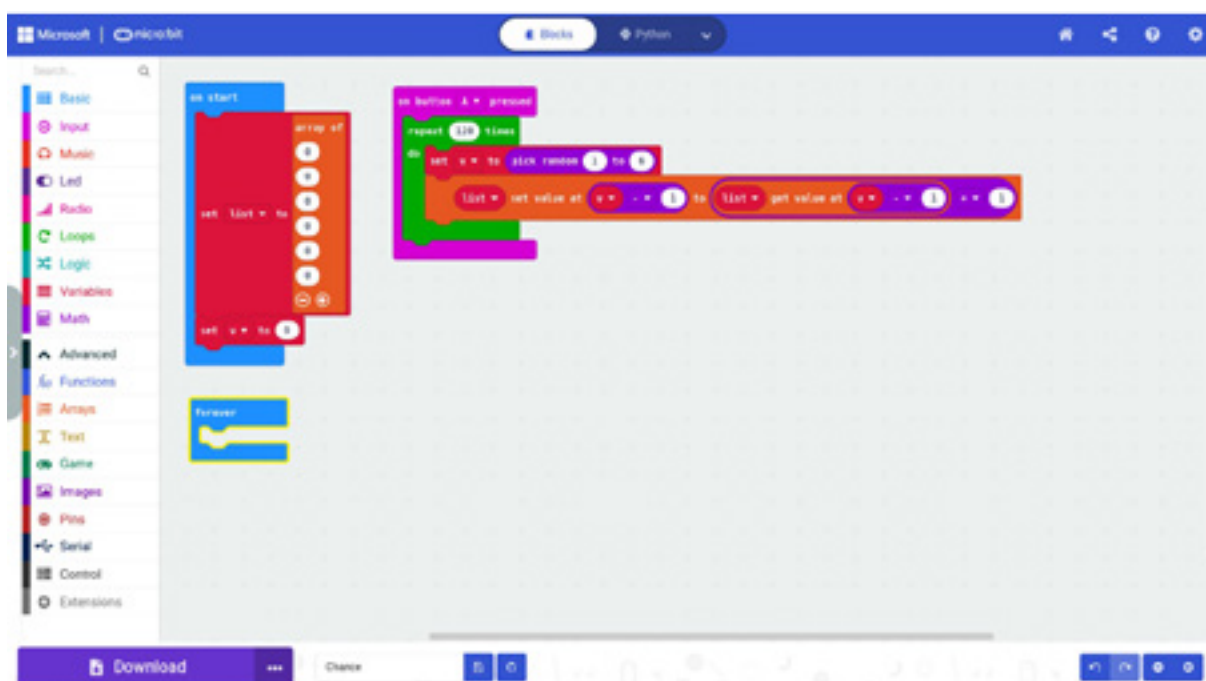
1. Otvori sekciju za nizove (*Advanced/Arrays*).



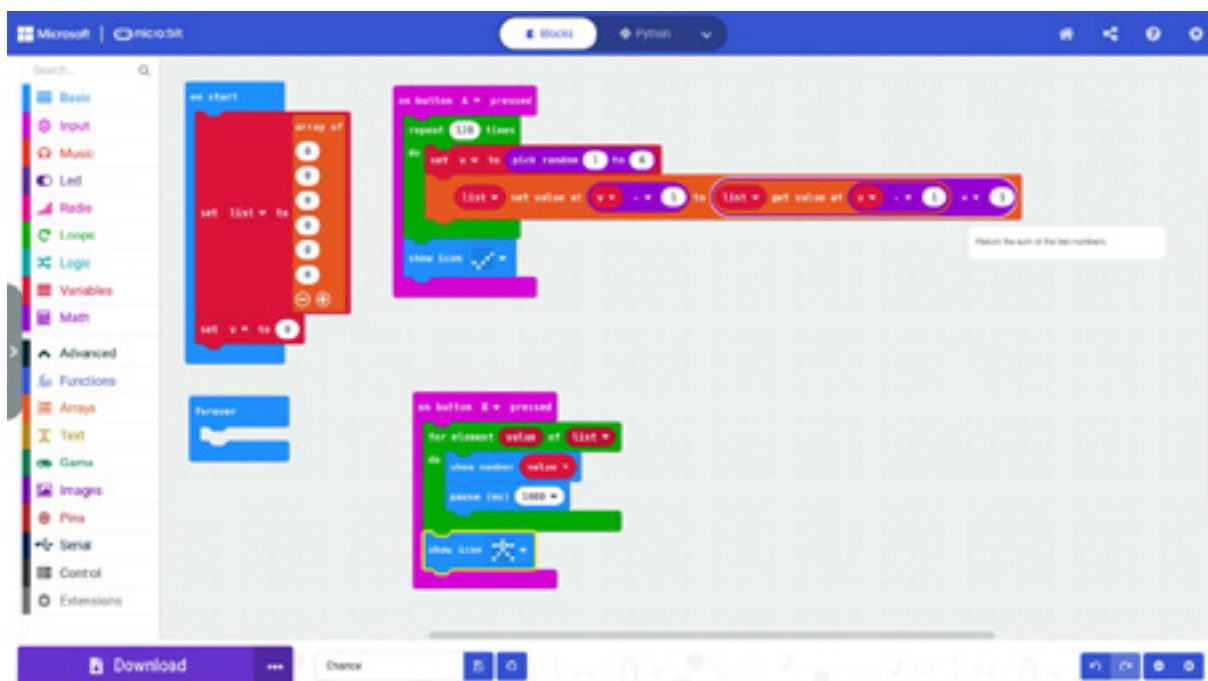
- Kreiraj listu od 6 brojeva tako da sve vrijednosti budu jednake 0. Kreiraj promjenljivu „v” sa početnom vrijednošću 0.



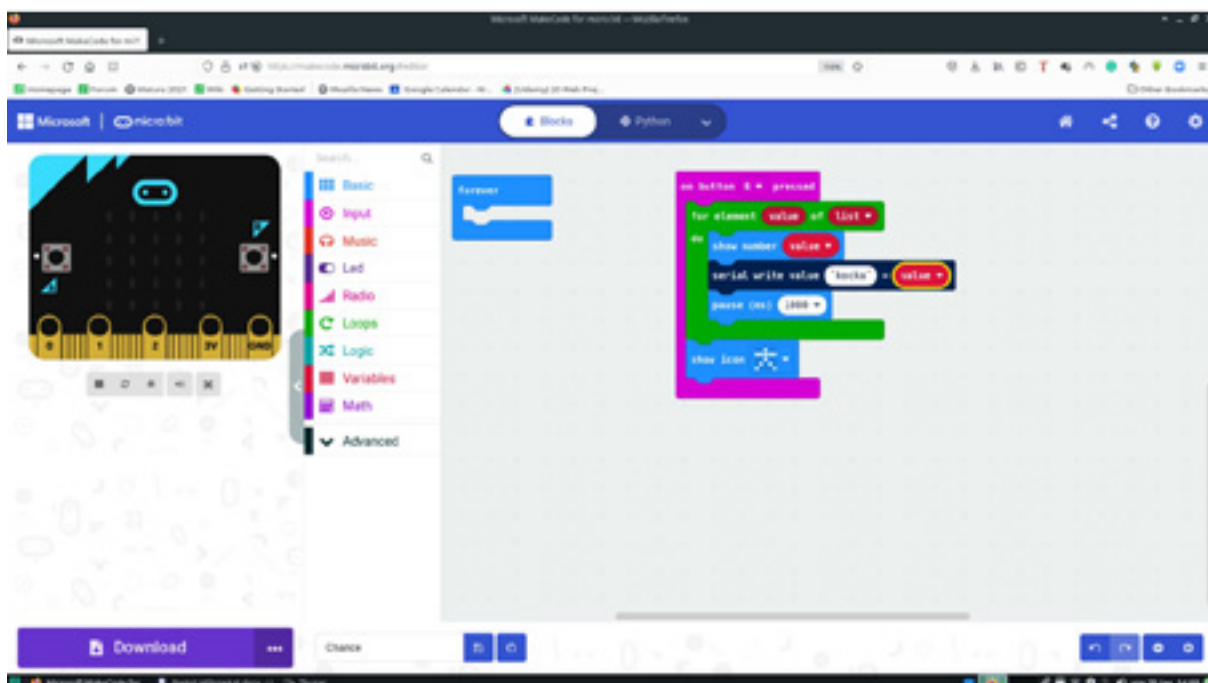
- Element sa indeksom *ind* prebrojava koliko se puta pri bacanju kockice dogodio ishod *ind* + 1. Na primjer, u list[0] upisujemo koliko je puta „pala” jedinica, dok u list[5] upisujemo koliko smo puta dobili šesticu. Dodaj blok *on button A pressed* i u njemu pomoću bloka *repeat ... times* ponavljaj proces bacanja kockice. Simulaciju jednog bacanja kockice ostvaruješ pomoću bloka *pick random 1 to 6*, a zatim uvećavaš odgovarajući element niza za 1 (tj. list[v-1] = list[v-1] + 1). Za izmjenu elemenata niza koristi blokove *list set value at ... to ...* i *list get value at ...*



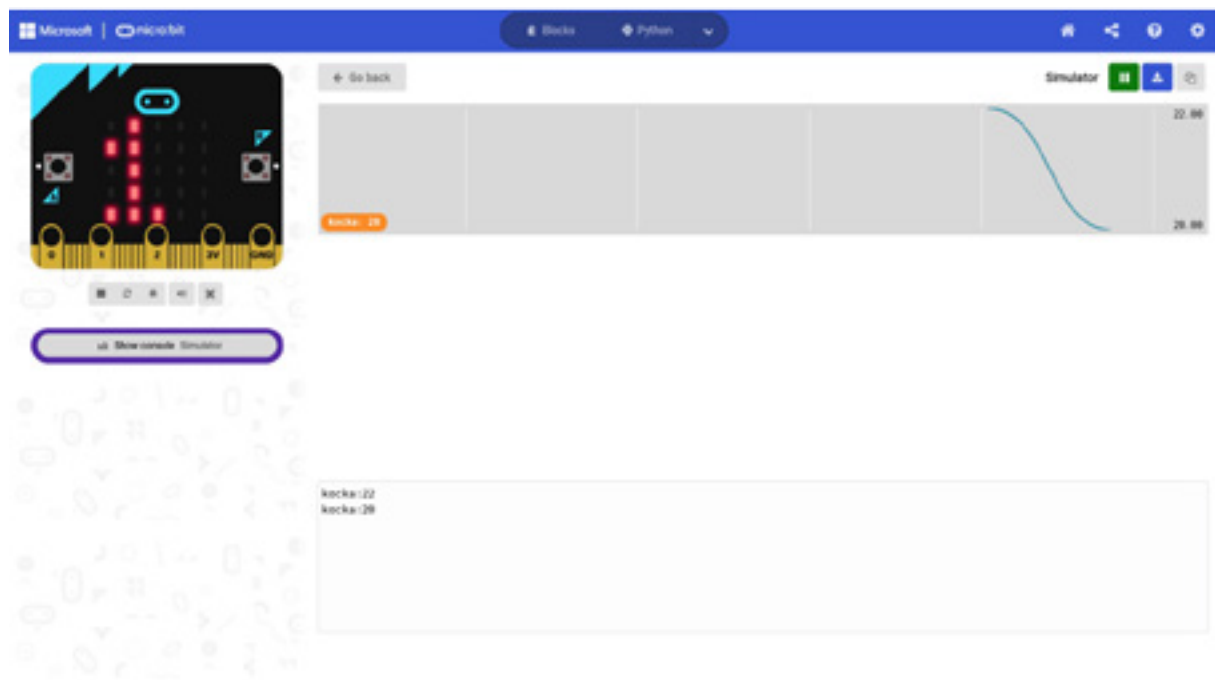
- U blok *on button B pressed* dodaj blokove za prikaz elemenata niza pomoću ciklusa *for element value of list*. U oba bloka *on button ...* dodaj ikonicu na kraju, koja će označiti kraj ciklusa.



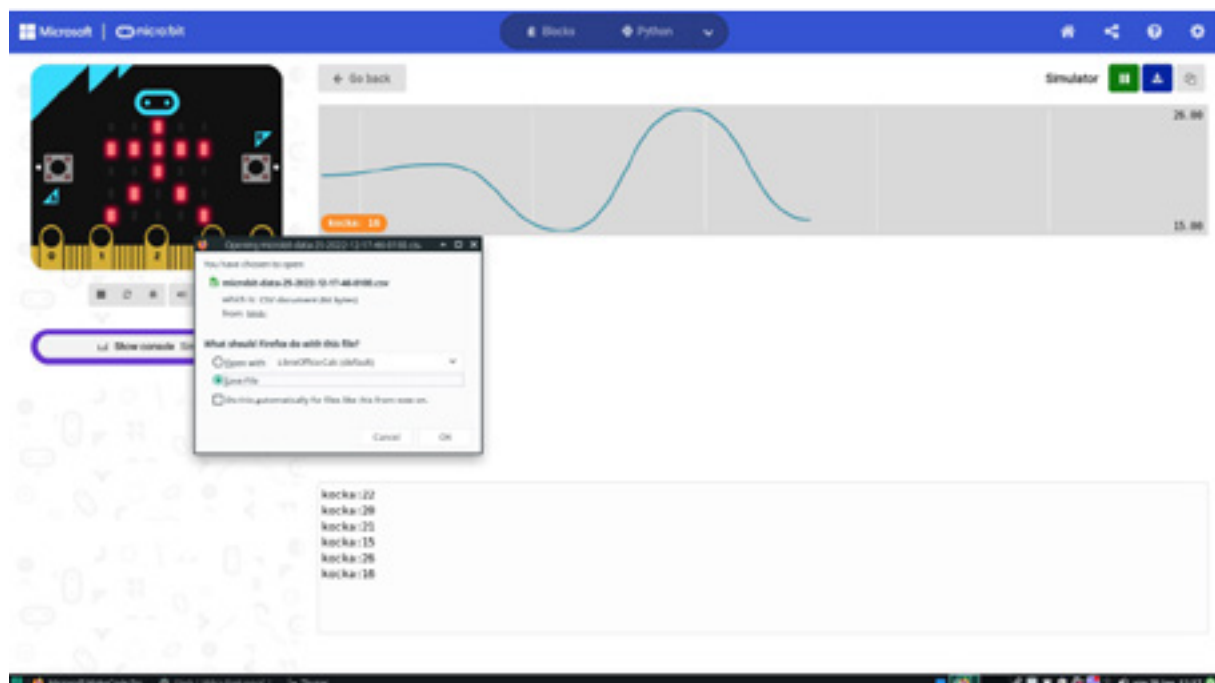
- U blok *on button B pressed* dodaj blok *serial write value kocka = value*.



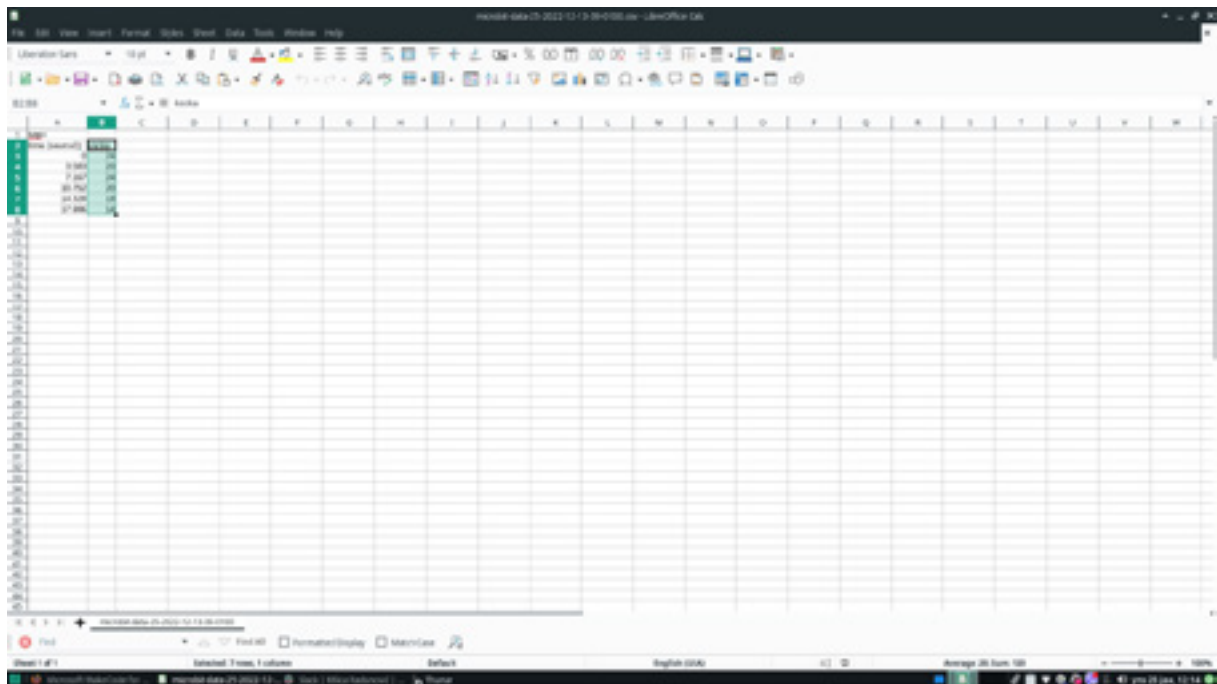
- Klikom na dugme A, generiši niz bacanja kockice, a zatim klikom na dugme B, prikaži dobijeni rezultat. Klikni na „show console Simulator”.



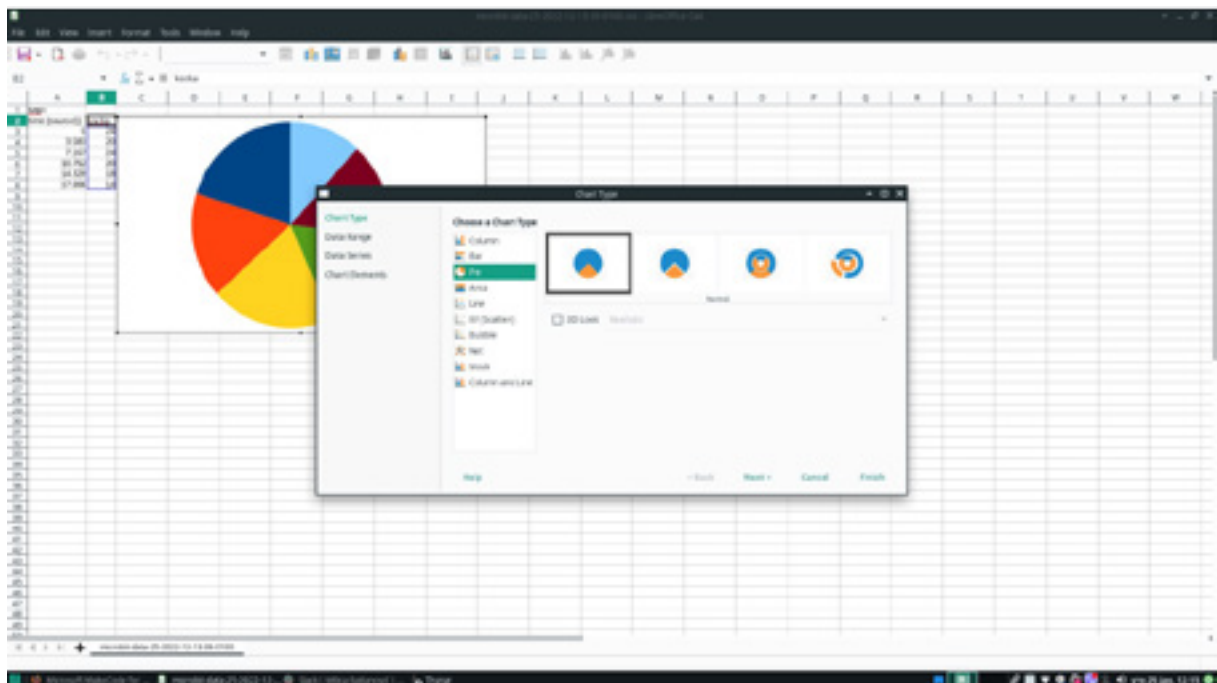
- Snimi rezultate izvršavanja programa u CSV fajl.



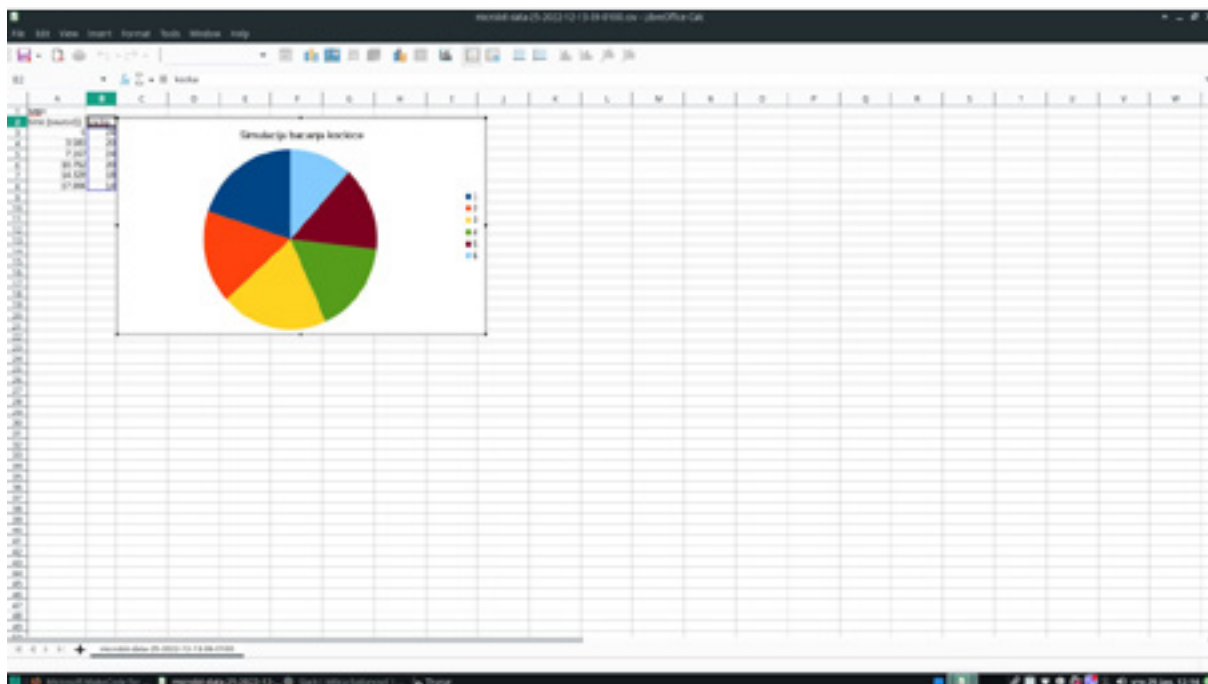
- Otvori dobijeni fajl u programu za tabelarne proračune (MS Excel, Libre Office Calc, Google Excel...) i označi polje Kocka i šest brojeva u koloni (na slici je prikazan izgled fajla pomoću programa Libre Office Calc).



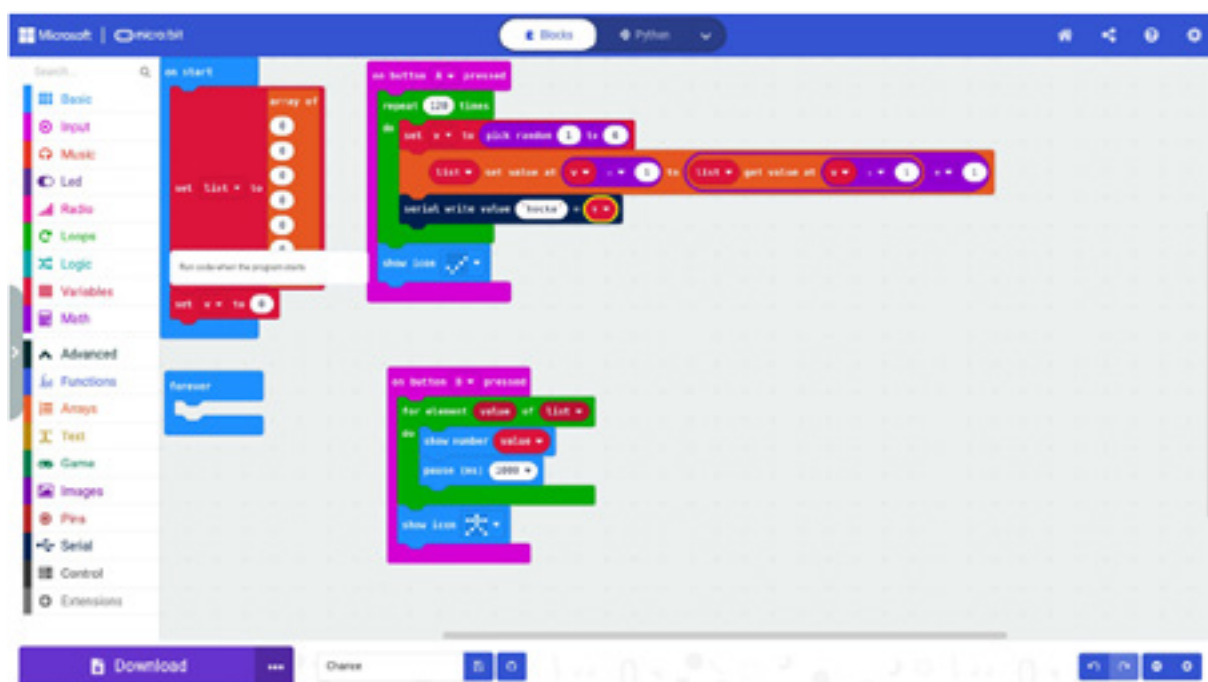
- Kreiraj grafikon oblika pite (engl. *pie*).



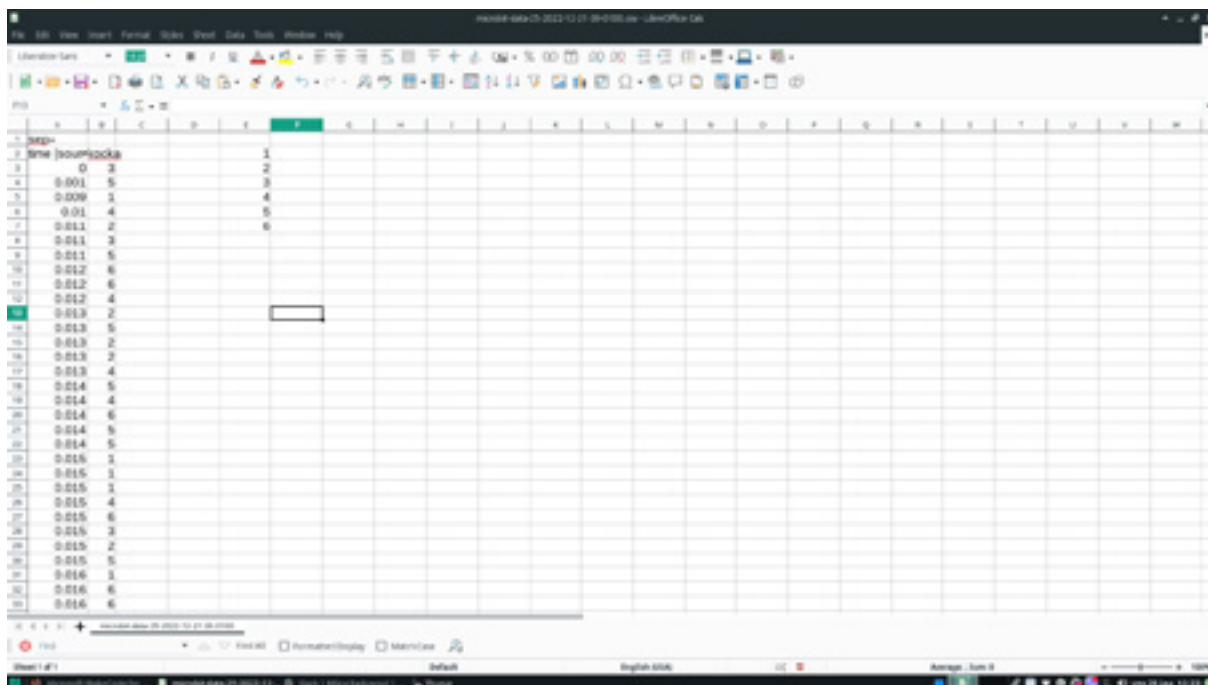
- Uredi grafikon tako da ima naslov „Simulacija bacanja kockice” i legendu sa desne strane grafikona.



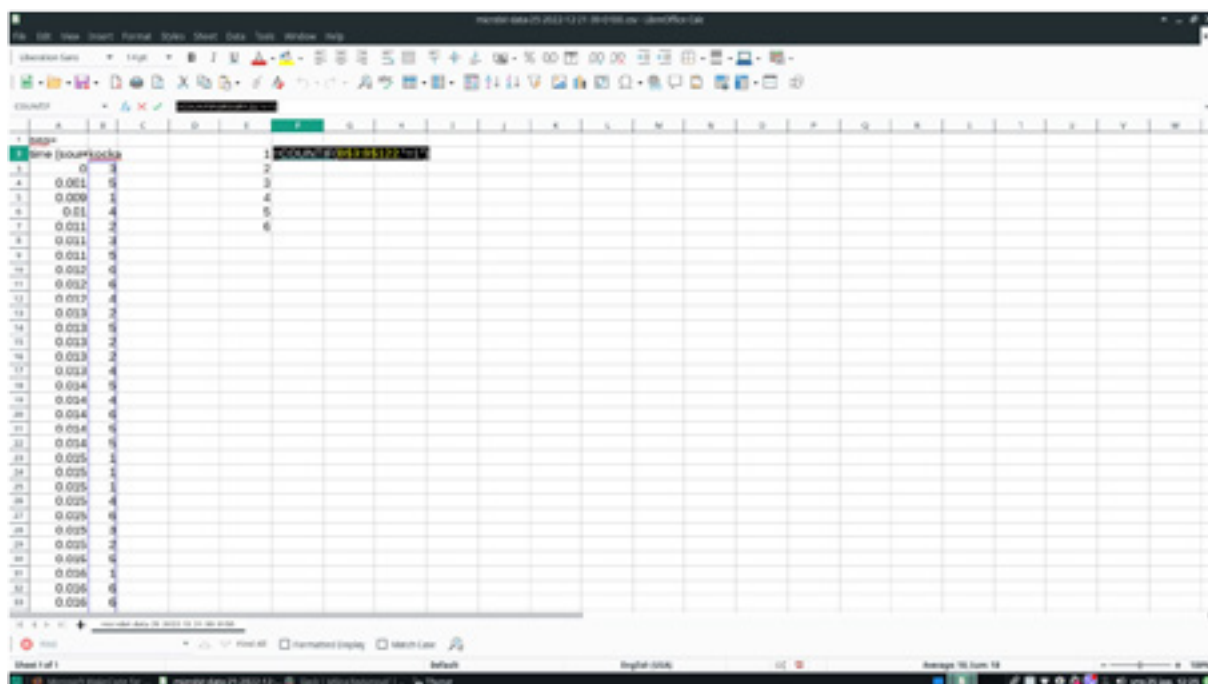
- Promijeni broj bacanja kockice u ciklusu *repeat* i ponovi korake 6–10. Uporedi grafikone iz svakog fajla i uoči da porastom broja ponavljanja, svaki grafikon sve više liči na podjelu kruga na 6 jednakih djelova.
- Iz bloka *on button B pressed* izbriši blok *serial write value ...* U blok *on button A pressed* dodaj blok *serial write value kocka = v*. Na ovaj način ćeš u fajl upisivati rezultate svakog bacanja kockice. Ponovi korake 6 i 7.



13. Otvori dobijeni fajl u programu za tabelarne proračune (MS Excel, Libre Office Calc, Google Excel...). Sada dobijamo mnogo veći broj redova, pa je potrebno da primjenom funkcije COUNTIF prebrojimo koliko je u našem eksperimentu dobijeno ishoda 1, 2, 3, 4, 5 i 6. U koloni E2:E7 dodamo brojeve 1, 2, 3, 4, 5 i 6.

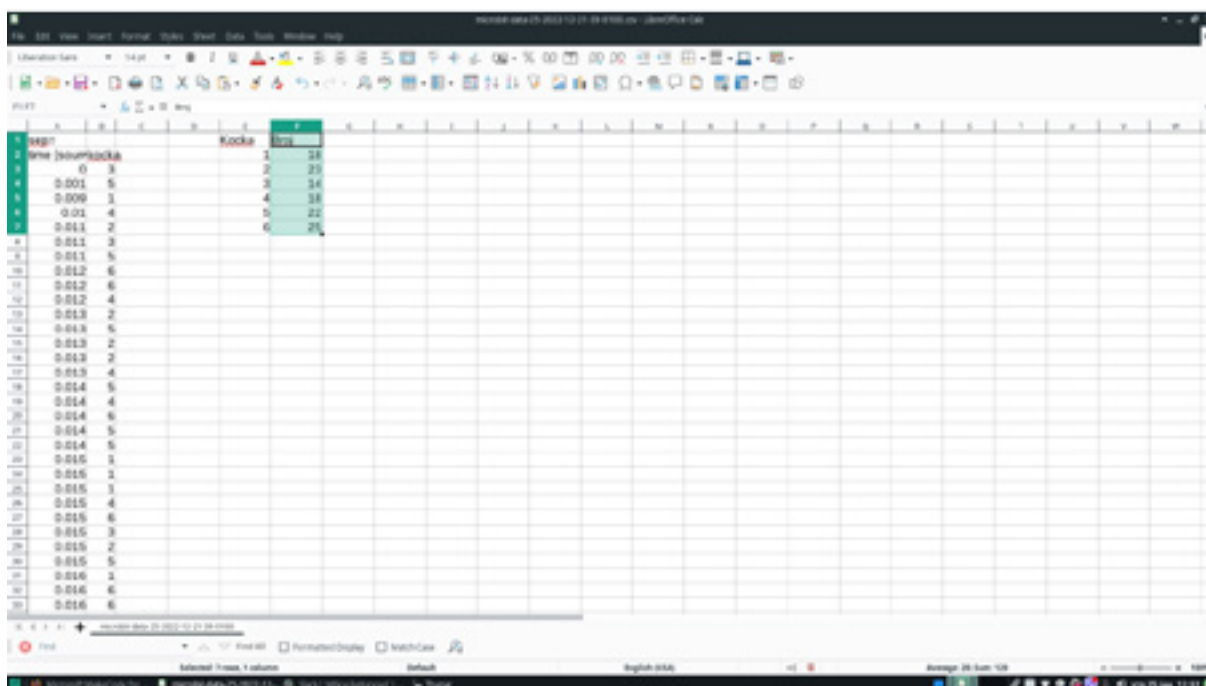


14. U eksperimentu je broj bacanja kockice 120, pa su rezultati raspoređeni u polja B3:B122. U polje F2 upišemo sljedeću formulu: =COUNTIF(\$B\$3:\$B\$122, "=1").



15. Ponovi ovaj postupak za polja F3:F7, tako da prebrojite koliko ste puta dobili ostale brojeve 2, 3, 4, 5 i 6. Formule koje treba upisati u polja date su u sljedećoj tabeli:

Polje	Formula
F2	=COUNTIF(\$B\$3:\$B\$122, "=1").
F3	=COUNTIF(\$B\$3:\$B\$122, "=2").
F4	=COUNTIF(\$B\$3:\$B\$122, "=3").
F5	=COUNTIF(\$B\$3:\$B\$122, "=4").
F6	=COUNTIF(\$B\$3:\$B\$122, "=5").
F7	=COUNTIF(\$B\$3:\$B\$122, "=6").



16. Mogući izgled fajla prikazan je na slici. Tvoji rezultati neće se u potpunosti poklopiti sa brojevima sa slike, jer se generišu na slučajan način.

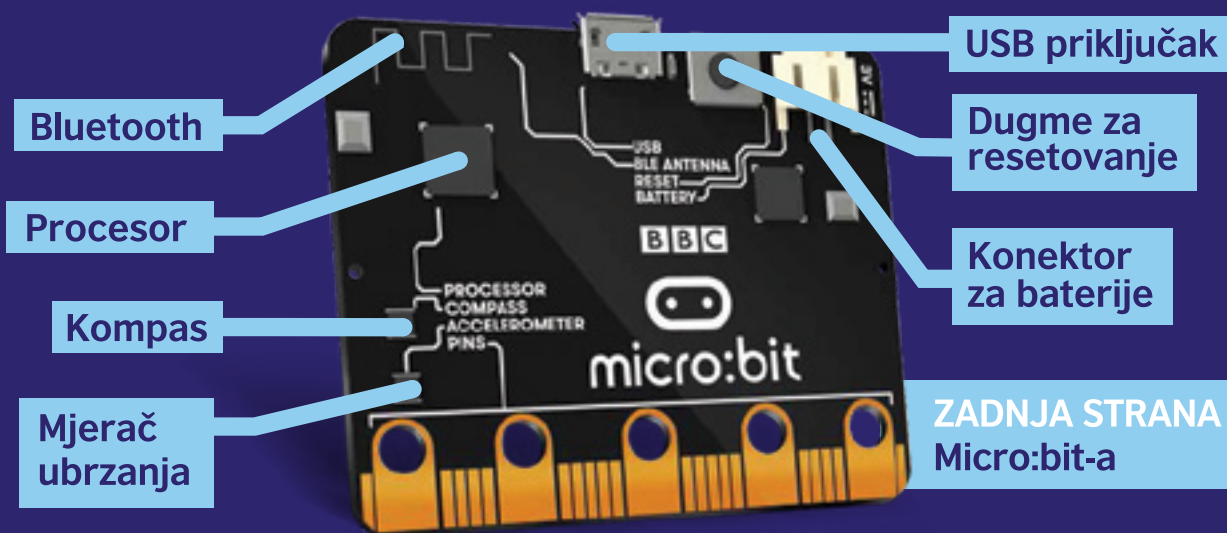
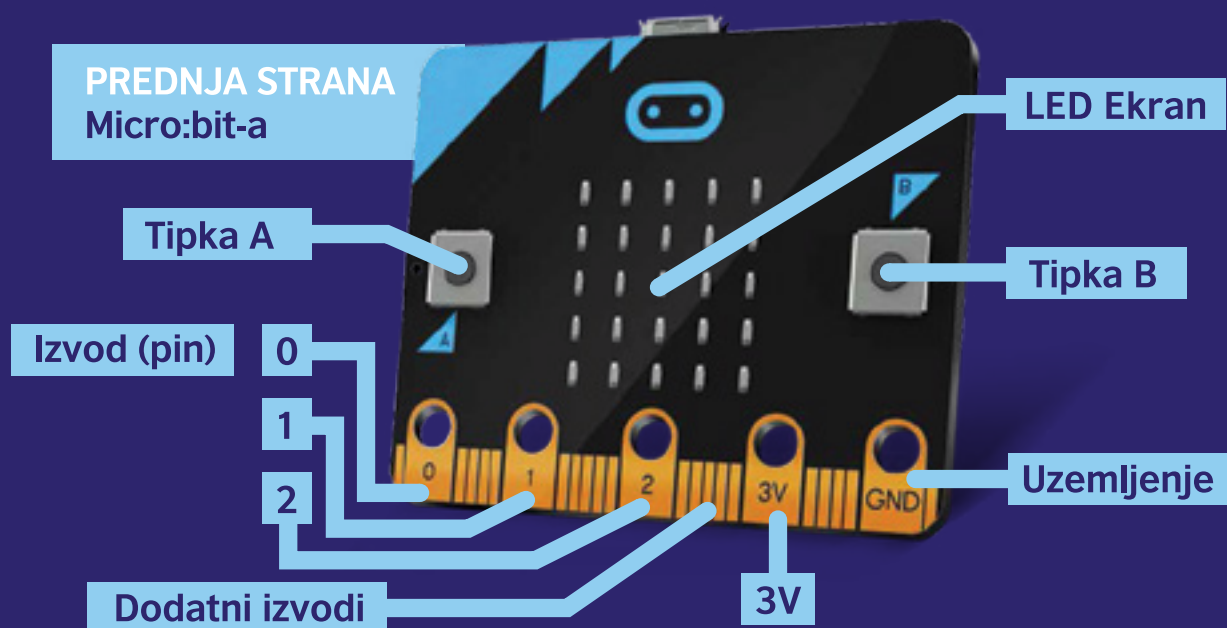
17. Ponovi korake 9, 10 i 11.

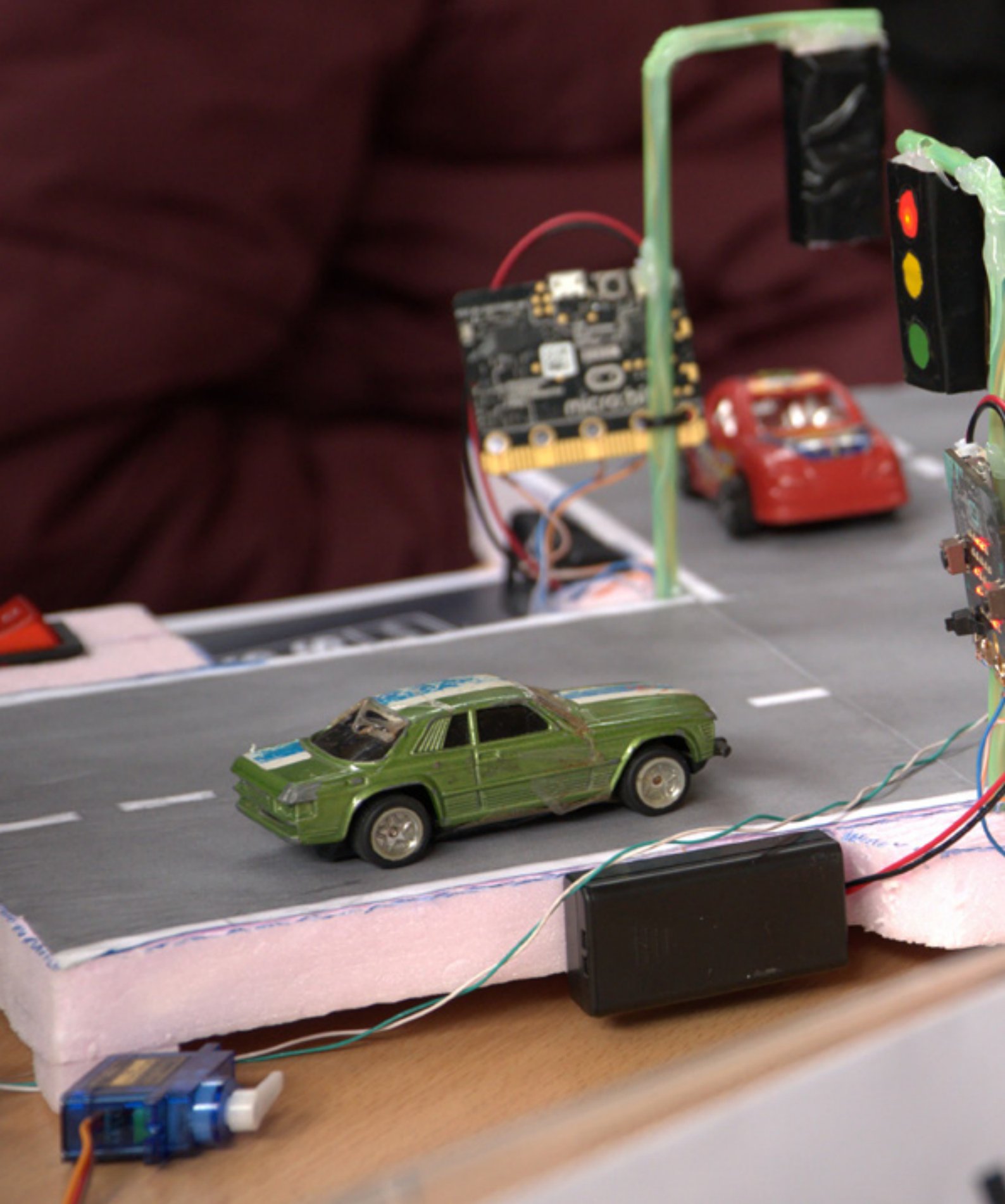
18. Prebaci program sa tvog računara na Micro:bit uređaj.

Program možete pogledati kada skenirate QR kôd ili na linku:

https://makecode.microbit.org/_1wu4v2icghgv







© British Council 2022

British Council je britanska međunarodna organizacija za kulturne odnose i obrazovanje. Djelujemo u preko 100 zemalja na poljima umjetnosti i kulture, engleskog jezika, obrazovanja i civilnog društva. U 2019. godini ostvarili smo kontakt sa 80 miliona ljudi direktno odnosno sa 791 milionom ljudi ukupno uključujući *online* platforme, TV i radio emisije i štampane publikacije. Dajemo pozitivni doprinos zemljama sa kojima sarađujemo, mijenjajući živote ljudi tako što stvaramo mogućnosti za razvoj, povezujemo ljude i institucije i gradimo povjerenje. Osnovani smo 1934. godine kao britansko neprofitno društvo koje se rukovodi Kraljevskom poveljom i istovremeno smo javna ustanova. Od britanske vlade dobijamo 15% finansiranja kroz osnovni grant.

www.britishcouncil.org

