BBC micro: bit i njegova primjena u razrednoj nastavi

Lesničar, Mirna

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:247466

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet za odgojne i obrazovne znanosti

MIRNA LESNIČAR

BBC micro:bit i njegova primjena u razrednoj nastavi

Diplomski rad

Pula, listopad 2018.

Sveučilište Jurja Dobrile u Puli
Fakultet za odgojne i obrazovne znanosti

MIRNA LESNIČAR

BBC micro:bit i njegova primjena u razrednoj nastavi

Diplomski rad

JMBAG: 0303039376, redoviti student
Studijski smjer: Integrirani preddiplomski i diplomski sveučilišni učiteljski studij

Predmet: Računalni praktikum
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: dr. sc. Ivan Pogarčić
Komentor: Janko Žufić, dipl. ing. el., v.pred.

Pula, listopad 2018.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Mirna Lesničar, kandidatkinja za magistru primarnog obrazovanja (mag. prim. educ.) ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, 2018. godine

IZJAVA

o korištenju autorskog djela

Ja, Mirna Lesničar dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom

BBC micro:bit i njegova primjena u razrednoj nastavi

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____2018.

Potpis

_____

1. Uvod 1
2. BBC micro:bit STEM revolucija u Hrvatskoj 2
3. Zaklada micro:bit 4
4. Micro:bit mini računalo 5
5. Upotreba micro:bit mini računala 7
6. Programiranje micro:bita 10
7. Uređivač za JavaScript Blocks 12
7.1. Kategorija naredbi 20
7.1.1. Basic 20
7.1.2. Input 23
7.1.3. Music 25
7.1.4. Led 28
7.1.5. Radio 29
7.1.6. Loops 29
7.1.7. Logic 30
7.1.8. Variables 31
7.1.9. Math 31
8. Uređivač za pisanje programa u Pythonu 34
9. Primjena BBC micro:bita u razrednoj nastavi 36
9.1. Matematika 36
9.2. Hrvatski jezik 46
9.3. Priroda i društvo 47
9.4. Likovna kultura 51
10. Zaključak 54
11. Literatura 55

Sažetak

Ovaj rad objašnjava primjenu micro:bit mini računala u razrednoj nastavi. Njegova korelacija je provedena kroz sve odgojne i obrazovne predmete od prvog do četvrtog razreda. Primjenom programiranja sa micro:bitom omogućavamo učenicima razvijanje aktivnog prostora za učenje te potičemo razvoj njihovih sposobnosti i znanja.

Summary

This study explains the application of micro:bit mini computers in primary school education. The correlation of micro:bit is implemented through all courses from first to fourth grade. By using programming with micro:bit, we enable all the students to develop their active space for studying and we foster the develpoment of their skills and knowledge.

1. Uvod

Kod objašnjavanja pojma „programiranje“ obično se kaže da je to način na koji „govorimo“ računalu što treba raditi. Programirati se može u određenom programskom jeziku određenom sintaksom. Svaki programski jezik ima svoj način korištenja i primjene. Srž programiranja nije u liniji kôda, nego u načinu razmišljanja. Čovjek mora biti u stanju zamisliti željeni program i rastaviti ga na segmente koje prikazuje pomoću naredbi. Ti segmenti čine cjelinu koja se naziva programom.

U ovom diplomskom radu objasnit ću primjenu mini računala micro:bit u razrednoj nastavi. Micro:bit je elektronički sklop nalik maloj pločici koja predstavlja mini računalo. Pomoću osobnog računala, micro:bit se može programirati. Učenici razredne nastave to mogu činiti pomoću blokova naredbi u Blocks uređivaču.

Cilj ovog diplomskog rada je prikazati primjenu micro:bita u razrednoj nastavi i to u različitim predmetima.

2. BBC micro:bit STEM revolucija u Hrvatskoj

Pojam STEM (eng. science, technology, engineering, mathematics) je sve poznatiji u Hrvatskoj i njezinom obrazovnom području. STEM je akronim koji povezuje znanost, tehnologiju, inženjerstvo i matematiku [1].

Tijekom godina se sve više primjećuje kako u STEM području nedostaje obrazovanih ljudi. Mladi nisu zainteresirani za navedena područja. Studije u tom području se podržavaju sa stipendijama i školarinama, ali je sam izbor toga vrlo nedostupan jer ga dobivaju samo najbolji studenti [1].

U siječnju 2017. godine pokrenuta je kampanja STEM revolucije koja je omogućila osnovnim i srednjim školama, knjižnicama i ustanovama koje rade s djecom masovno uvođenje BBC micro:bit tehnologije. Kampanja pripada pod projekt Croatian Makers udruge IRIM (Institut za Razvoj i Inovativnost Mladih) kojemu je cilj digitalna razvijenost i pismenost mladih u Republici Hrvatskoj [2].

Ovu tehnologiju su razvili BBC (engl. *British Broadcasting Corporation*), Microsoft i drugi partneri sa ciljem da se 'besplatna' tehnologija brzo proširi i samim time poveća interes mladih osoba koje nisu upoznate sa ovim područjima. Njezina primjena nije samo u programiranju nego i u dizajnu, umjetnosti i ostalim predmetima [2].

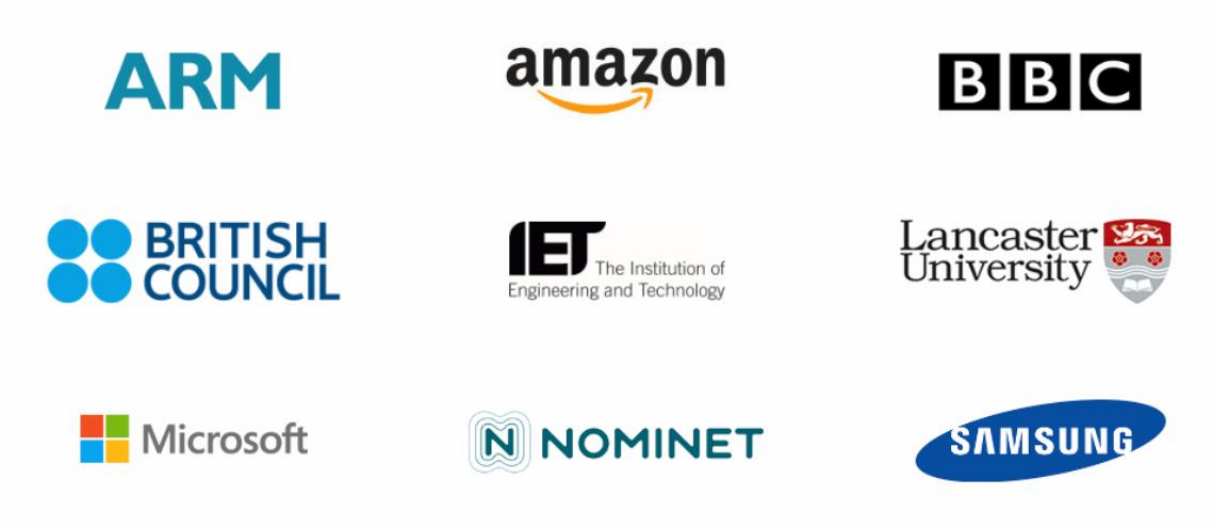
Jednostavost pristupa tom sadržaju je bežično spajanje i veliki broj senzora koji omogućavaju korisniku primjenu u svim osnovnoškolskim i srednjoškolskim predmetima kao što su kemija, biologija, glazbena kultura, informatika ili likovna kultura, uz naravno obavezne kurikulumske programe iz matematike, hrvatskog jezika te prirode i društva. Jednostavnost je u tome što primjenom BBC micro:bita djeca ulaze u digitalni svijet, a već iskusni programeri, inženjeri i znanstvenici ga mogu koristiti kao moćan alat u izvedbi projekata i zadataka njihovih ideja [2]. Isto tako, ova tehnologija je našla svoj put u izvannastavnim aktivnostima i na fakultetu.

Većina dostupnog materijala uputa za korištenje micro:bit alata je na engleskom jeziku. *Institut za razvoj i inovativnost mladih* (IRIM) iz Hrvatske proizveo je originalne materijale, a njihova dostupnost se nalazi na obrazovnoj web stranici: www.croatiansmakers.hr . IRIM se bavi uvođenjem STEM revolucije u Hrvatsku, a

njihov najveći projekt je *Croatian Makers* koji omogućava korisnicima pristup edukativnim materijalima i primjerima programa koji su izrađeni za korištenje BBC micro:bit alata. Osnivači udruge su Nenad i Rujana Bakić [20]. Isto tako, osnovan je forum na kojem svi učitelji mogu otvoreno komunicirati ili dijeliti materijale sa ostalim članovima.

3. Zaklada micro:bit

Ova neprofitna organizacija registrirana je u rujnu 2016. godine. Njezino sjedište se nalazi u Velikoj Britaniji čija je osnovna djelatnost donacija micro:bitova obrazovnim ustanovama širom svijeta. Njihova se službena stranica nalazi na web adresi www.microbit.org. Partneri zaklade pružaju podršku kroz volontiranja, nagrade i razvijanje novih tehnologija [3]. Svrha zaklade je uvođenje djece u digitalni svijet, a poseban naglasak stavljaju na žensku populaciju i na skupine djece u nepovoljnom položaju. Samim time, stvara se velika zajednica učitelja koja međusobno surađuje i izrađuje obrazovne materijale te pruža potporu korisnicima. Micro:bit je osmišljen na način da se počne primjenjivati od prvog dana nastave u školama za sve učenike od prvog razreda, za sve nastavne predmete. Prikupljene povratne informacije od učenika i učitelja su vrlo pozitivne. Porastao je interes za učenjem računarstva, a time se može očekivati i porast zanimanja za STEM područja u bliskoj budućnosti. Neke od država koje se koriste ovom tehnologijom su Finska, Šri Lanka, Singapur i SAD [4].

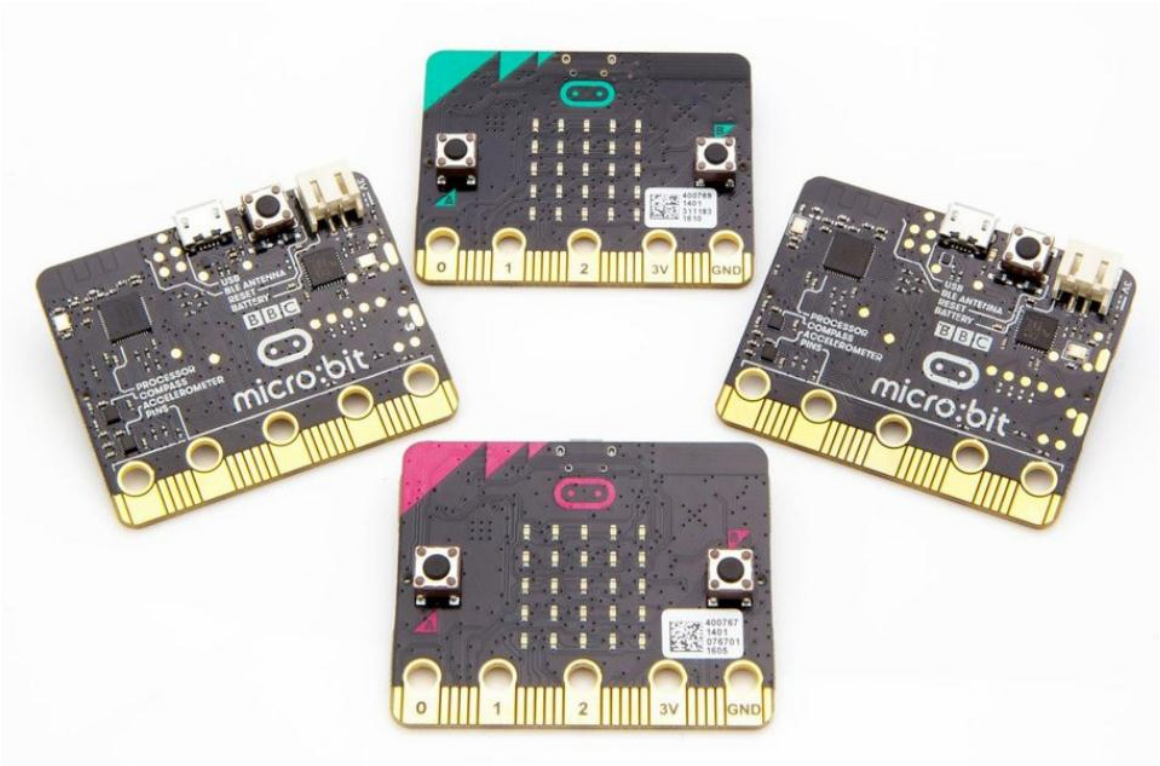


Slika 1. Osnivači zaklade BBC micro:bit

4. Micro:bit mini računalo

Micro:bit je mala pločica koja predstavlja mini računalo. Ovo se mini računalo može programirati. Mikro:bit sa mikrokontrolerom i senzorima omogućava programiranje i omogućava razvijanje logičkog razmišljanja prilikom 'sastavljanja' kodova programa. Pločica se programira putem online uređivača (eng. editora) na klasičnom računalu, a zatim se program putem USB sučelja prebaci na uređaj. Mikro:bit, osim senzora ima i dvije tipke i 5x5 LED matricu (Slika 2.) kao i 25 pinova koji čine elektroničko sučelje. Specifikacije mikro:bit računala su [5]:

- Mikrokontroler: 32-bit ARM Cortex M0+
- Senzori: Acelerometar, žiroskop, kompas
- Aktuatori: dva tipkala, LED matrica 5x5
- Komunikacija: Bluetooth, USB
- Konektori: JST za bateriju, tri generalna konektora te napajanje
- Dimenzije: 43mm x 52mm
- Sadržaj pakiranja: micro:bit mini-računalo, USB kabel, držač za baterije i baterije



Slika 2. Micro:bit mini računalo

Slika 3. Mikrokontroler: 32-bit ARM Crtex M0+

Micro:bit mini računalo je vrlo jednostavno i praktično. Izgled je primijenjen najmlađim korisnicima, a kompleksnost programa koji se mogu izvesti je neograničena. Može se primijeniti u mnogim područjima kao što su umjetnost, računarstvo, likovna i glazbena kultura, informatika te mnogi drugi predmeti od osnovnoškolskih do visokoškolskih obrazovnih ustanova.

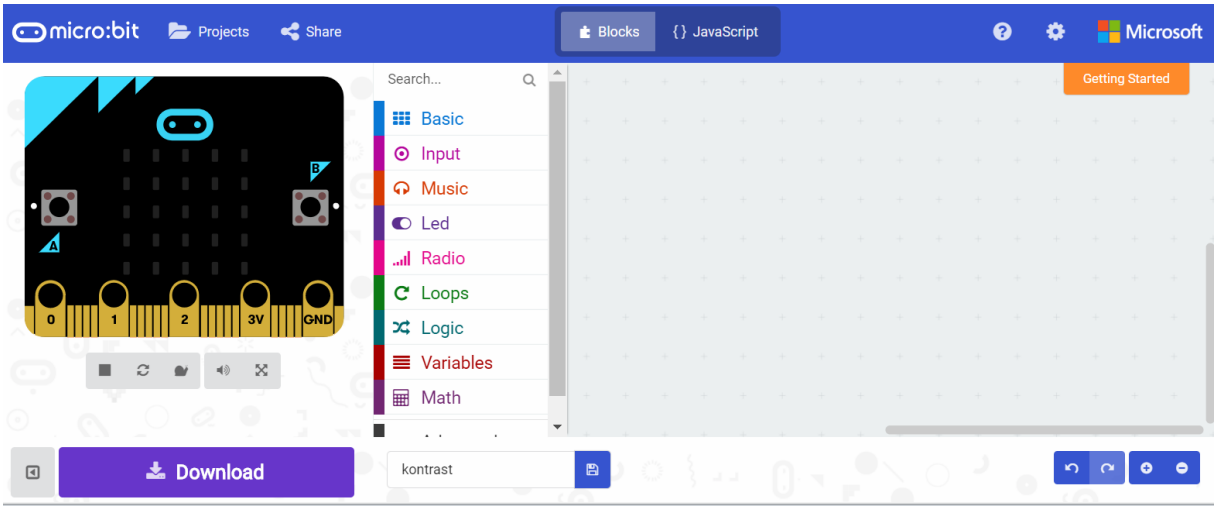
U Hrvatskoj micro:bit još nije dovoljno prepoznatljiv, iako su sve škole koje su željele, kao donaciju, dobile komplet od 20 micro:bita. U širenju prepoznatljivosti, u širenju micro:bita sudjeluje IRIM i Carnet. Volonteri i drugi obrazovni djelatnici kroz zakladu BBC micro:bit svakim danom predstavljaju nove mogućnosti primjene mikro:bitova. Rad na micro:bitu je prvi korak u upoznavanju djeteta s logičkim razmišljanjem te s programiranjem.

5. Upotreba micro:bit mini računala

Da bi učenici mogli koristiti micro:bit potrebno je upoznati ih s osnovama funkcioniranja micro:bita. Programiranje se provodi na besplatnom online uređivaču (engl. editor) na osobnom računalu.

Nakon što učenik napiše program, na simulatoru može provjeriti kako će program raditi na micro:bitu. Program na micro:bit može prenijeti pomoću USB sučelja i kabla. Micro:bit podržava prijenos podataka sa operacijskih sustava Linux, iOS i Raspaberry Pi [8].

Dakle, ukoliko smo se odlučili na programiranje putem online uređivača, potrebno je otići na web stranicu micro:bita (www.makecode.microbit.org). Otvoriti će se simulator i uređivač pomoću kojeg je omogućen početak programiranja (Slika 4).



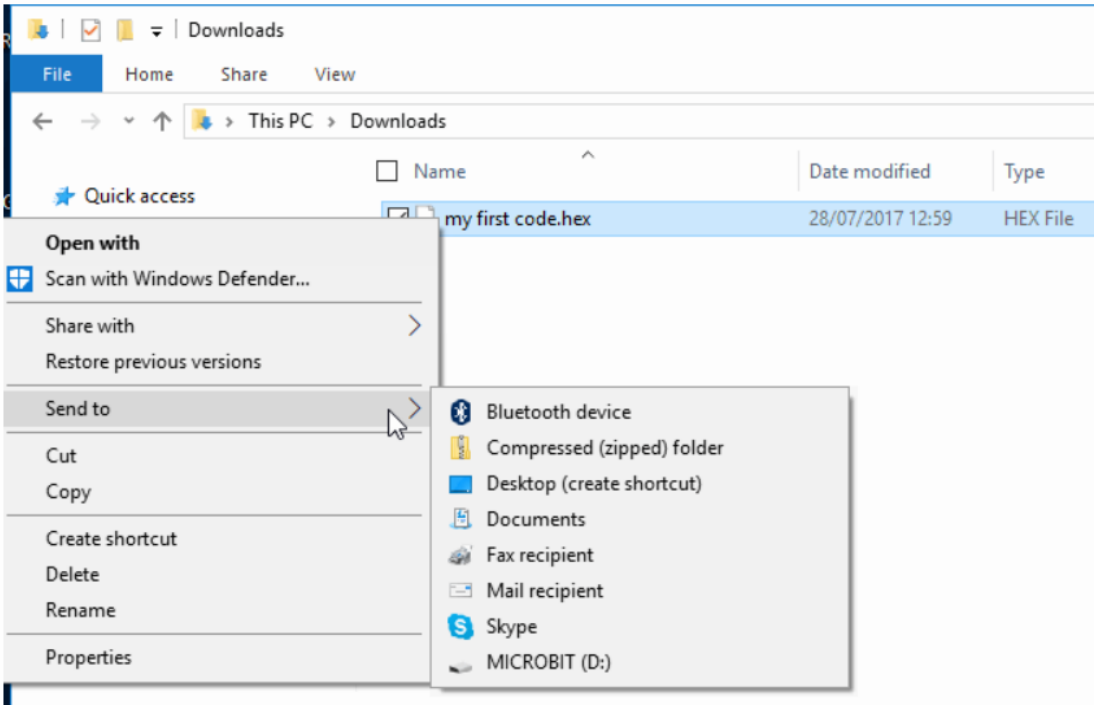
Slika 4. Sučelje micro:bit uređivača

Tehnikom 'drag and drop' omogućeno je korištenje naredbi u micro:bit simulatoru koji se sastoji od blokova naredbi (engl. Blocks). Nakon što je korisnik odabrao željenu naredbu, lijevom tipkom miša klikne na tu naredbu i prenosi ju pokazivačem miša na blok naredbi u kojem ju želi smjestiti. Otpuštanjem lijevog klika miša, naredba ostaje na mjestu pokazivača. Metoda 'drag and drop' je vrlo praktična tijekom brzog odabira i primjene željene naredbe. Podloga blokova napisana je u programskim jezicima PHP i JavaScriptu.

Nakon što je korisnik „napisao“ u uređivaču na osobnom računalu i odlučio spremiti svoj program na osobnom računalu, u donjem dijelu programskog sučelja nalazi se

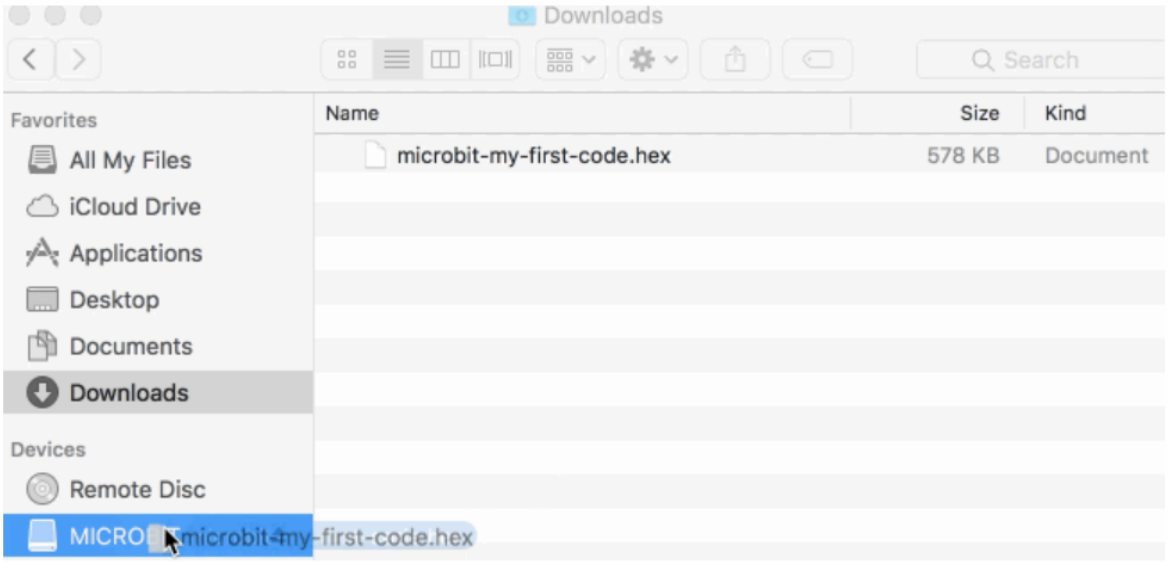
opcija 'Download' koja omogućava spremanje programa na micro:bit. Format u kojem se datoteka sprema je '.hex' format [8].

Kako većina korisnika ima Windows operacijski sustav, korisnicima je omogućeno olakšano prebacivanje programa na micro:bit mini računalo. Desnim klikom miša na datoteku, odabiremo naredbu 'send to – MICROBIT' (Slika 5).



Slika 5. Način prebacivanja datoteke na micro:bit

Korisnici iMac operativnog sustava koriste metodu 'drag and drop' kako bi prenijeli datoteku s osobnog računala na micro:bit mini-računalo (Slika 6).



Slika 6. Način prebacivanja datoteke sa iMac OS

Nakon što je korisnik prebacio program na micro:bit mini-računalo, pali se žuta LED dioda koja korisniku signalizira da se program učitava. Nakon kompletnog prebacivanja datoteke, program se automatski pokreće.

U micro:bit mini-računalu u isto vrijeme može biti pohranjen samo jedan program i to samo u .hex formatu. Nakon učitavanja druge datoteke sa osobnog računala, dosadašnja se datoteka zamjenjuje novom [8].

6. Programiranje micro:bita

Kako bismo uopće pristupili izradi programa za micro:bit, moramo znati na koji način se ta vrsta programiranja provodi. Tijekom rada korisnik se sa svojim osobnim računalom spaja s micro:bitom. Prilikom toga, korisnikovo računalo može biti na Internetu i koristiti uređivač (eng. editor) ili može koristiti uređivač instaliran na računalo. Uređivače se može podijeliti na:

- Vizualne blok editore i
- Tekstualne editore

Vizualni blok editori (hrv. uređivači) su jednostavniji za korištenje jer se sastoje od blokova naredbi pomoću kojih se stvara program. Njegovu funkcionalnost možemo prikazati kao igru s kockicama od kojih se kreira složeniji program. Blokovi s naredbama predstavljaju kockice koje se sastoje od različitih boja. Te boje predstavljaju kategorije prema kojima se dijele naredbe. Uz svaku naredbu možemo povezati točno određene naredbe kako bi kôd funkcionirao. Na taj način je nemoguće koristiti krivu naredbu u kodu, jer se 'kockice' neće posložiti. Neki od najpoznatijih editora su: Microsoft Block Editor, Microsoft PXT, Microsoft Touch Develop [6], JavaScript Blocks i drugi.

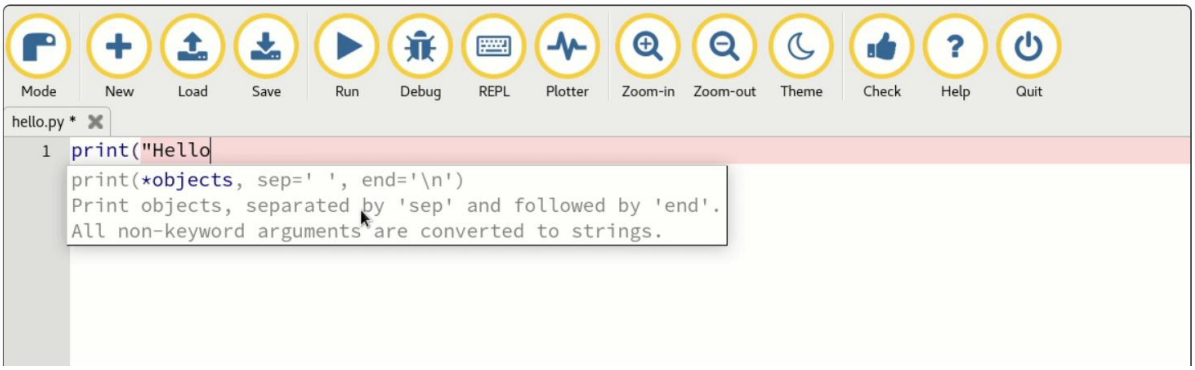
Što se tiče online **tekstualnih editora**, oni se od standardnih programa razlikuju po sučelju. Za njihovu primjenu potreban je veći nivo znanja programiranja. Njihova specifičnost je ručni unos kôda, a samim time i poznavanje programskog jezika koji je potreban za određeni kod. Neki od tih jezika mogu biti: C, C++, Python, JavaScript i mnoštvo drugih.

Komponenta koja spaja tekstualni i vizualni blok editor je Microsoft Touch Develop. To je platforma koja omogućava spajanje programa i programskih jezika preko kojih korisnik može programirati i uređivati kod prema svojim zamislima. Njegov rad je predviđen za interakciju s pametnim telefonom ili tabletom jer omogućava korisniku prijenos programa na micro:bit putem Bluetooth tehnologije [7].

Kao što smo spomenuli korisnik može tijekom rada biti spojen na internet, a i ne mora. Offline programiranje se odnosi na programe koji se nalaze na našem računalu. Najbolje rješenje koje je povezano sa izradom koda u micro:bit-u je MU

editor (Slika 7). On omogućava pretvaranje u .hex izvedbenu datoteku koju koristi sam micro:bit. Programiranje se na taj način vrši u microPython-u.

Početni koraci u korištenju ovog [4] programskog jezika su složeniji u usporedbi sa Blocks JavaScript-om u online uređivaču micro:bita, tako da se može koristiti u višim razredima osnovne škole ili sa naprednijim učenicima razredne nastave.



Slika 7. Programsko sučelje MU

Osim MU editora postoji i Visual Studio Code koji služi za pisanje kodova na nekoliko različitih programskih jezika (Slika 8). Način rada mu je isti kao kod MU editora. Korisnik može birati u kojem programskom jeziku želi prikazati svoje kôdove, a nakon toga ih pretvara u .hex format, pogodan za micro:bit mini računalo, koji se sprema na osobno računalo te potom šalje na micro:bit mini-računalo.

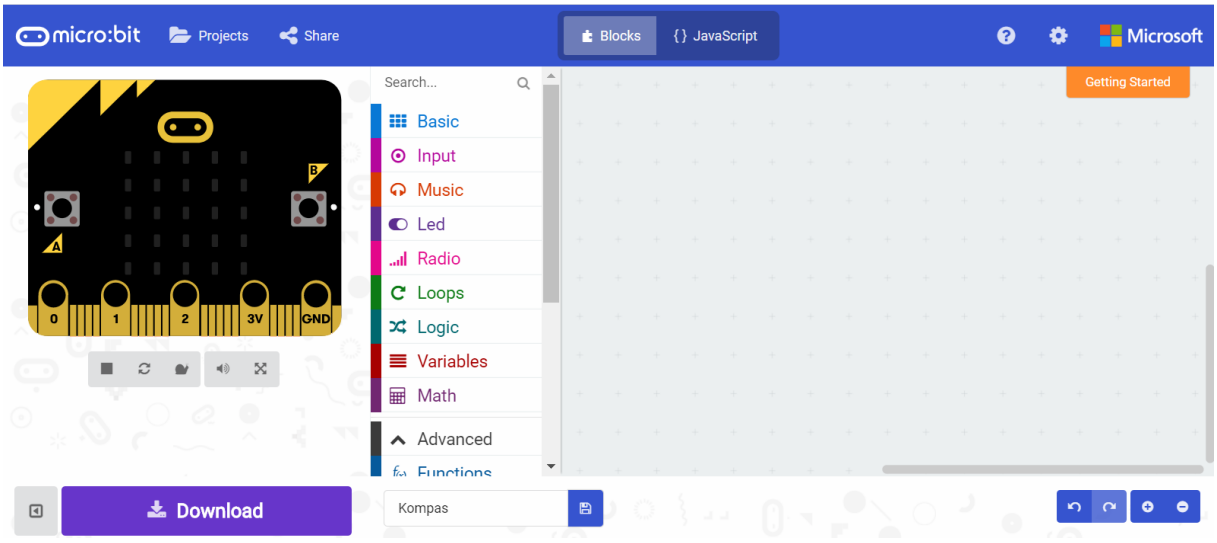
Slika 8. Visual Studio Code

7. Uređivač za JavaScript Blocks

JavaScript Blocks uređivač je vizualno programsko okruženje koje omogućava korisniku programiranje putem blokova ili JavaScript načina pisanja programskog kôda [9]. Kao što smo ranije napomenuli on pripada skupini vizualnih blok uređivača te je zato prikladan u radu s učenicima razredne nastave. Iskusniji korisnici mogu pisati kôd u JavaScriptu koji spada u tekstualne uređivače [6].

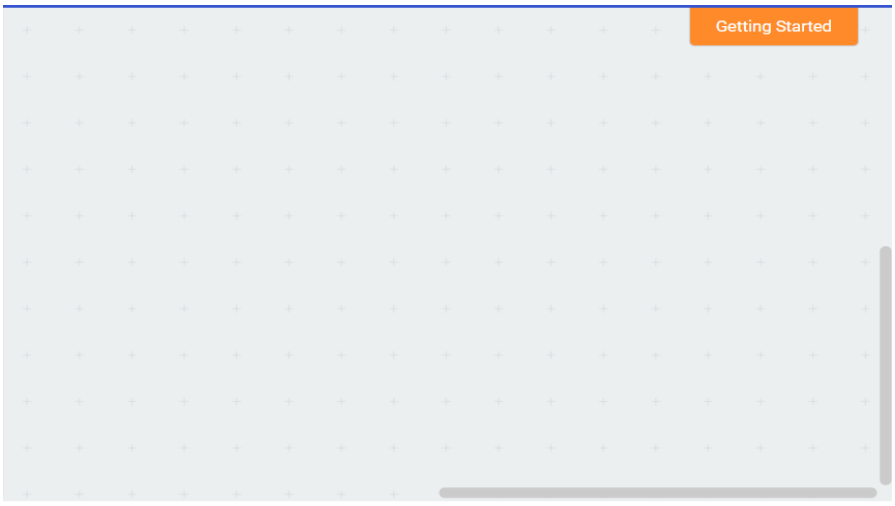
Naredbe u JavaScript Blocks uređivaču su podijeljene u kategorije (Slika 9). Kategorije naredbi razlikuju se po bojama kako bi se na brz i učinkovit način mogla pronaći željena vrsta naredbi. Sam uređivač sastoji se od:

- Prostora za slaganje kodova
- Kategorija naredbi
- Simulatora
- Aktivacijskih gumbova za određivanje načina kodiranja
- Tipki za snimanje gotovog programa
- Aktivacijskog gumba za otvaranje projekta
- Aktivacijskog gumba za dijeljenje programa
- Postavki i poveznica



Slika 9. Uređivač za JavaScript Blocks

Prostor za slaganje kodova odnosi se na komponiranje željenog programa spajanjem bloka naredbi u funkcionalnu cjelinu (Slika 10).

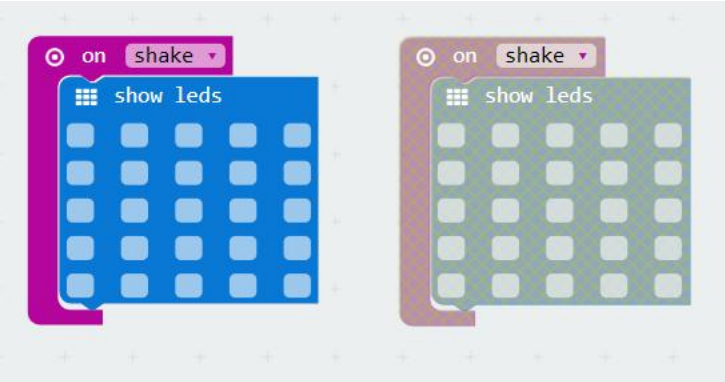


Slika 10. Prostor za slaganje kodova

Prilikom 'slaganja kodova' korisniku je omogućeno povećanje i smanjenje prikaza radne površine programa, a isto tako postoji i tipka za poništavanje prethodnog koraka (engl. Undo) ili vraćanje istog korakom unaprijed (Slika 11).

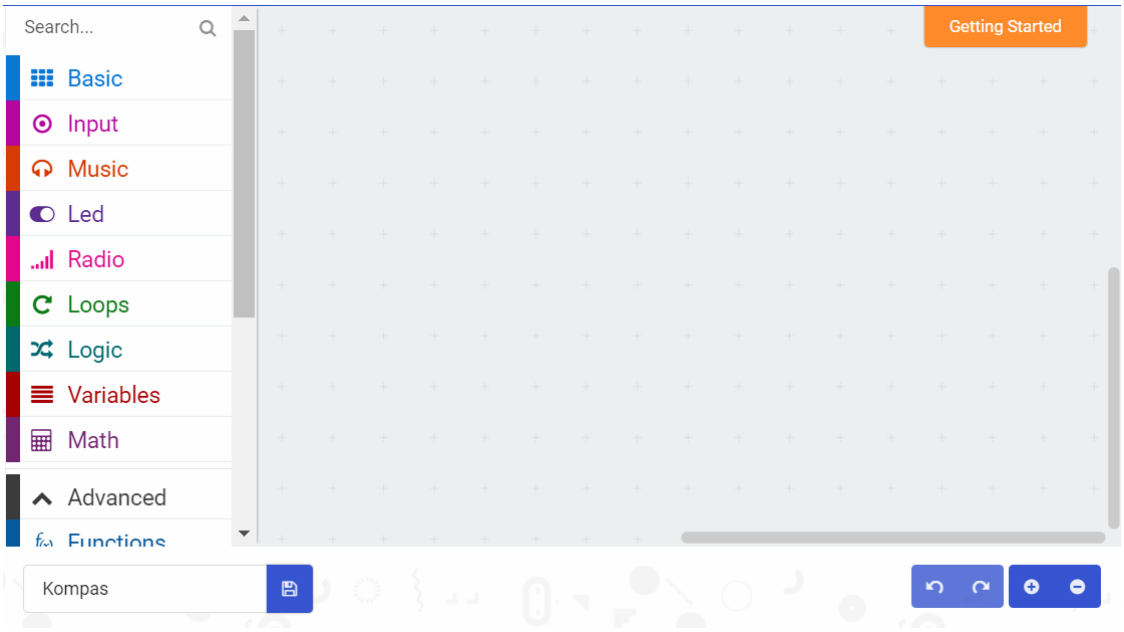
Slika 11. Tipke za poništavanje koraka unaprijed i unazad

Nakon što je korisnik odabrao željenu blok naredbu, metodom 'drag and drop' postavlja ju na željeno mjesto u prostor za izradu koda. Ključna stvar u izradi programa je da svaki blok naredbi mora odgovarati drugom. Prilikom uspješnog spajanja blokova naredbi, čuti će se 'klik'. To je znak da odabrane naredbe funkcioniraju. Ako se spoje dvije krive naredbe, promijeniti će se boja odabrane kategorije te ona izgleda 'prozirno'. Razlika između kompatibilnog i nekompatibilnog bloka naredbi može se vidjeti na slici 12.



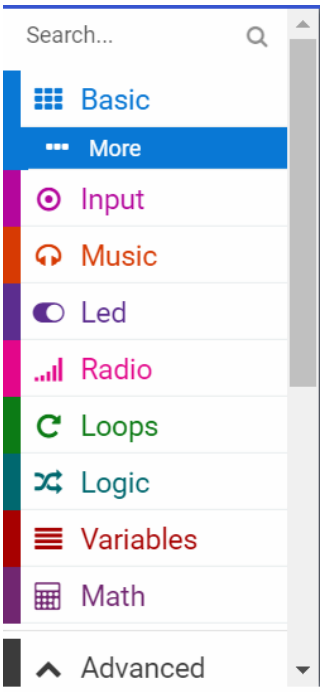
Slika 12. Primjer ispravnog (jasna slika) i neispravnog (slika je osjenčana) postavljanja blokova naredbi

Nadalje, blok koji sadrži **kategorije naredbi** nalazi se u središnjem dijelu programa te je svaka kategorija označena različitom bojom kao i naredbe koje se u njoj nalaze (Slika 13). Sve kategorije nisu vidljive odjednom, nego se mogu pronaći korištenjem klizne trake sa desne strane bloka kategorija naredbi.



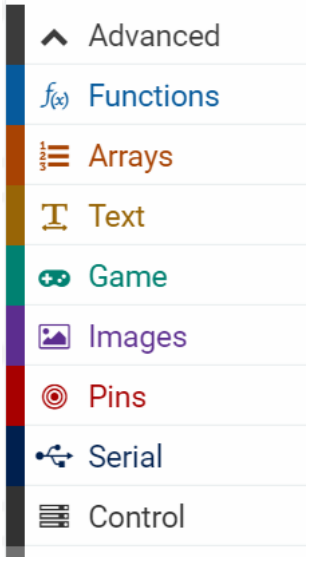
Slika 13. Kategorije naredbi u različitim bojama

Za svaku kategoriju postoje napredne opcije (Slika 14), do njih dolazimo odabirom određene kategorije i lijevim klikom miša na istu. Nakon toga se otvara podprogram '..More' koji se sastoji od dodatnih naredbi te kategorije.



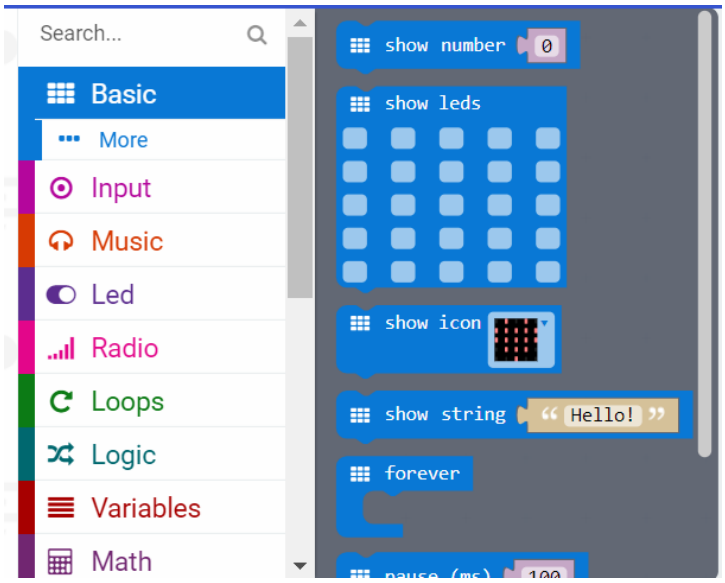
Slika 14. Opcija 'More' za kategoriju naredbi Basic

Isto tako, postoji grupa naredbi pod nazivom 'Advanced' koja se sastoji od naprednijih kategorija kao što su matematičke funkcije, tekst, igre ili slike (Slika 15). Njihova primjena će ovisiti o nivou znanja koje je korisnik usvojio prilkom programiranja u micro:bitu.



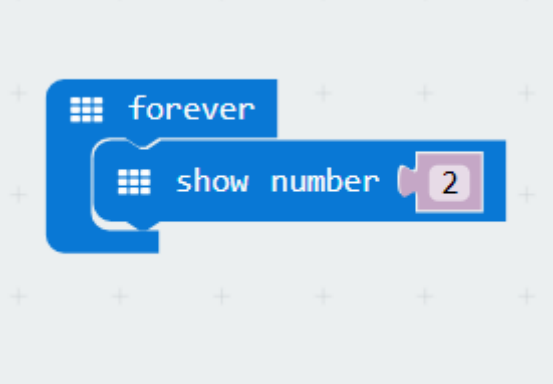
Slika 15. Grupa naredbi 'Advanced'

Odabirom željene kategorije otvara se novi izbornik u kojem se nalaze naredbe koje spadaju u tu kategoriju. Metodom 'drag and drop' korisnik odvlači naredbu u prostor za kreiranje kôda (Slika 16). Na primjer, kad korisnik odabere kategoriju naredbi 'Basic' i želi odabrati naredbu 'Show number' tada izabere i povuče naredbu na prostor za kreiranje kôda. Unosom broja stvara se program će na simulatoru prikazati taj broj.



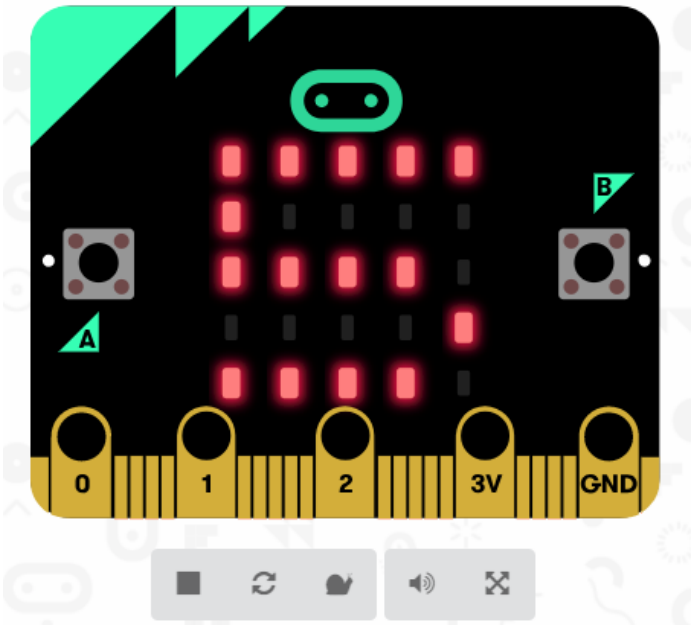
Slika 16. Primjer odabiranja naredbe 'show number'

Nakon unesene naredbe 'Show number' ona se nalazi u prostoru za kreiranje koda i na nju se nadovezuje naredba koja određuje hoće li se zadani broj prikazivati neprestano kad program radi ili će određenom aktivacijskom tipkom njegova prisutnost biti ograničena (Slika 17).



Slika 17. Izgled naredbe 'Show number' u prostoru za pisanje koda

Kako bismo provjerili ispravnost unesenih naredbi možemo pogledati trenutni prikaz programa koji izrađujemo. Taj prikaz se ostvaruje kroz **simulator**. On se nalazi s lijeve strane uređivača i on je aktivan prilikom dodavanja novih naredbi programu. Korisniku je na taj način omogućen pregled programa prije nego ga odluči prebaciti na micro:bit mini-računalo. Ispod samog simulatora nalaze se gumbi za zaustavljanje i ponovno pokretanje (engl. restart) programa, usporeni prikaz izvedbe programa (engl. Slow Mo) te gumb za određivanje jačine zvuka i prikaz preko cijelog ekrana (Slika 18).



Slika 18. Prikaz simulatora na osobnom računalu

Ukoliko u kodu koristimo određene naredbe koje zahtjevaju fizičko spajanje na vanjske elemente ili uređaje, simulator to ne može prikazati. Isto tako, aktivacijski gumb 'Show the simulator' nam omogućuje sakrivanje simulatora i proširivanje prostora za izradu koda.

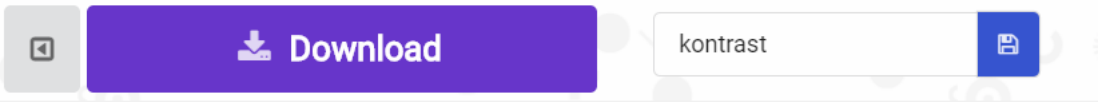
Prije nego krenemo sa izradom programa, moramo odlučiti hoćemo li koristiti vizualni ili tekstualni uređivač. Tekstualni uređivač nam omogućava pisanje koda u JavaScriptu, a vizualni uređivač se sastoji od gotovih blokova naredbi koje su objašnjene.

U gornjem dijelu uređivača za JavaScript Blocks nalaze se **aktivacijski gumbovi za određivanje načina kodiranja** putem kojih odabiremo željeni način programiranja pritiskom na jedan od izbornika (Slika 19).

Slika 19. Sktivacijski gumbovi za određivanje načina kodiranja

Za početnike je idealno započeti programiranje uz 'Blocks' uređivač jer nudi gotove blokove s naredbom uz koje svaki korisnik razvija logičko razmišljanje i usmjerava svoj rad ozbiljnom programiranju. Izvrstan je za mlađe polaznike kao što su učenici osnovnih škola, a moglo bi se reći i djeca predškolske dobi. Povezivanjem naredbi i određivanjem kategorija, dolazi do efikasnog rješenja za funkcioniranje zamišljenog programa. Što se tiče pisanja kôdova JavaScript, korisnik mora poznavati naredbe i sintaksu programskog jezika JavaScript.

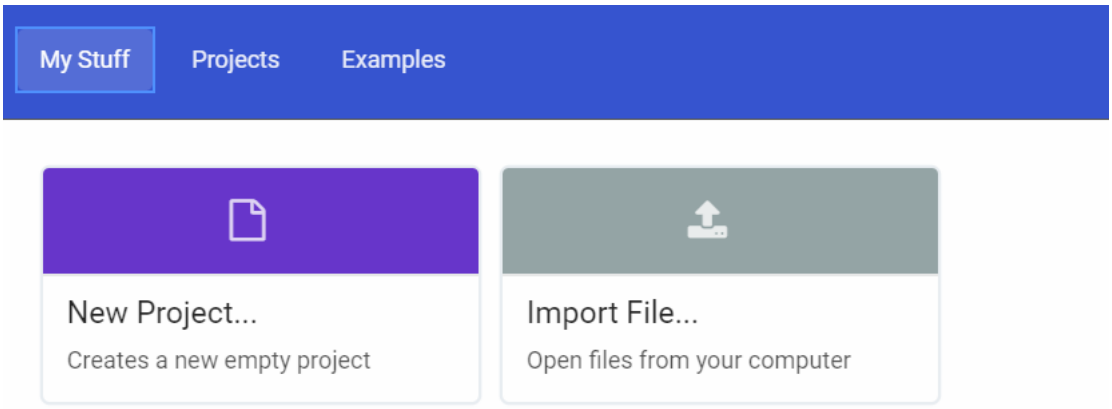
Svaki program kojeg korisnik odluči prebaciti na micro:bit mini-računalo ili zadržati na osobnom računalu, može to učiniti pomoću tipki za snimanje gotovog programa. One se nalaze u donjem kutu uređivača. Uz **tipke za spremanje programa** (engl. Download) nalazi se prostor za dodavanje imena programa. Ukoliko korisnik ne unese ime programa, uređivač automatski sprema skinuti program pod nazivom 'untitled', a to otežava korisniku pronalazak spremljenog programa na osobnom računalu. Nakon što korisnik odredi ime programa, preuzimanje .hex datoteke se vrši klikom na gumb 'Download' ili na gumb sa sličicom diskete (Slika 20).



Slika 20. Preuzimanje micro:bit programa

Ako želimo podijeliti svoje programe sa ostalim korisnicima, to je moguće putem **aktivacijskih gumbova za dijeljenje programa** ili projekata. U bilo kojem trenutku korisniku je omogućeno dijeljenje programa sa ostatkom svijeta.

Odabirom na gumb '**Projects**' otvara se izbornik '**My stuff**' u kojem možemo odabrati želimo li otvoriti novi prazni projekt ili želimo otvoriti jedan od projekata koji se već nalaze na našem računalu. Isto tako, vidljiv je prikaz kad smo zadnji put pristupili nekom od programa koje smo radili u online uređivaču (Slika 21).

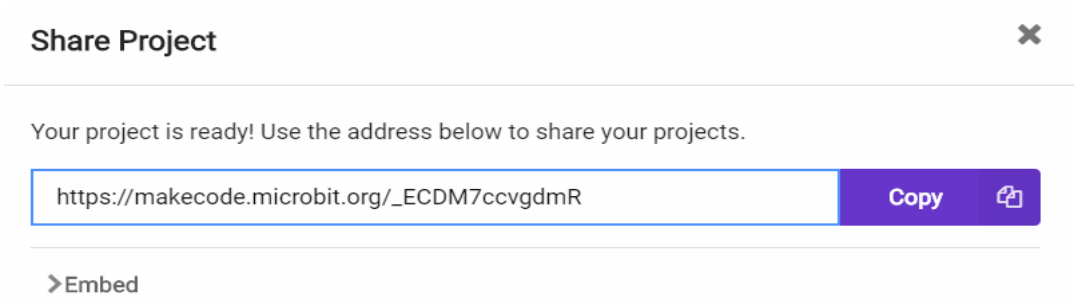


Slika 21. Otvaranje novog projekta

Sljedeća kartica kojoj možemo pristupiti je kartica '**Projects**' u kojoj možemo odabrati projekt koji sami izrađujemo prateći upute koje se prikazuju u prostoru za izradu programa. Što se tiče kartice '**Examples**' u njoj se nalaze gotovi programi koje možemo preuzeti i koristiti na našoj pločici.

Nakon što smo izradili projekt. Možemo ga podijeliti sa ostalim korisnicima. To ćemo uraditi odabirom gumba 'Share' u lijevom gornjem dijelu online uređivača. Prije samog dijeljenja projekta, prvo moramo kliknuti na gumb 'Publish project' te nakon toga biramo na koji ćemo način podijeliti svoj projekt.

Ukoliko želimo dijeliti samo trenutno otvoreni program u Blocks uređivaču, možemo kopirati poveznicu na kojoj se on sprema (Slika 22).



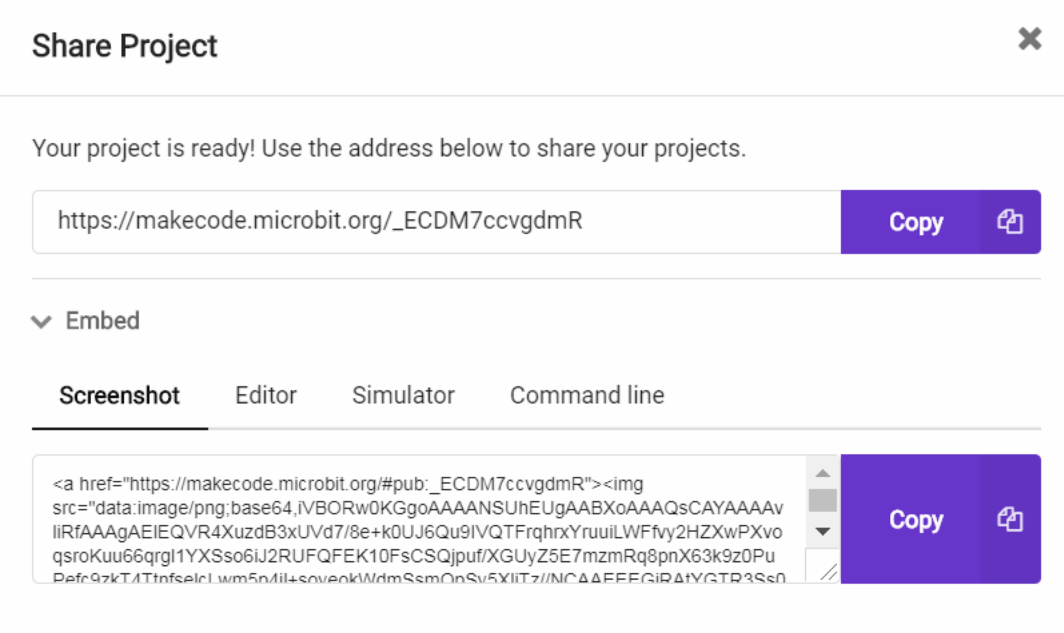
Slika 22. Dijeljenje projekta

Naredbom 'Embed' otvaramo klizni prozor u kojem su nam ponuđene četiri opcije (Slika 23). Projekt koji smo izradili možemo:

- Prikazati u obliku slike zaslone (engl. Screenshot)
- Mijenjati u uređivaču (engl. Editor)

- Animacija (engl. Simulator)
- HTML kod kao dio web stranice (engl. Command line)

Odabirom bilo koje od ovih naredbi otvara se linijski kod kojeg kopiramo i prenosimo na željenu stranicu ili unutar određenog programskog jezika.



Slika 23. Prikaz dodatnih opcija u dijeljenju projekta

7.1. Kategorija naredbi

Blok naredbi sa kategorijama glavni je dio online uređivača. Iz ovih kategorija se naredbe preuzimaju i povlače u dio za kreiranje kôda, a zatim spajaju i nastaju gotovi programi. Oni su podijeljeni po skupina, a svaka skupina iz kategorije je označena drugom bojom. Na taj način je korisniku omogućeno lakše snalaželje s naredbama i njihovim spajanjem. Ukoliko želimo spojiti naredbu iz jedne kategorije, s naredbom iz druge, to ponekad neće biti moguće.

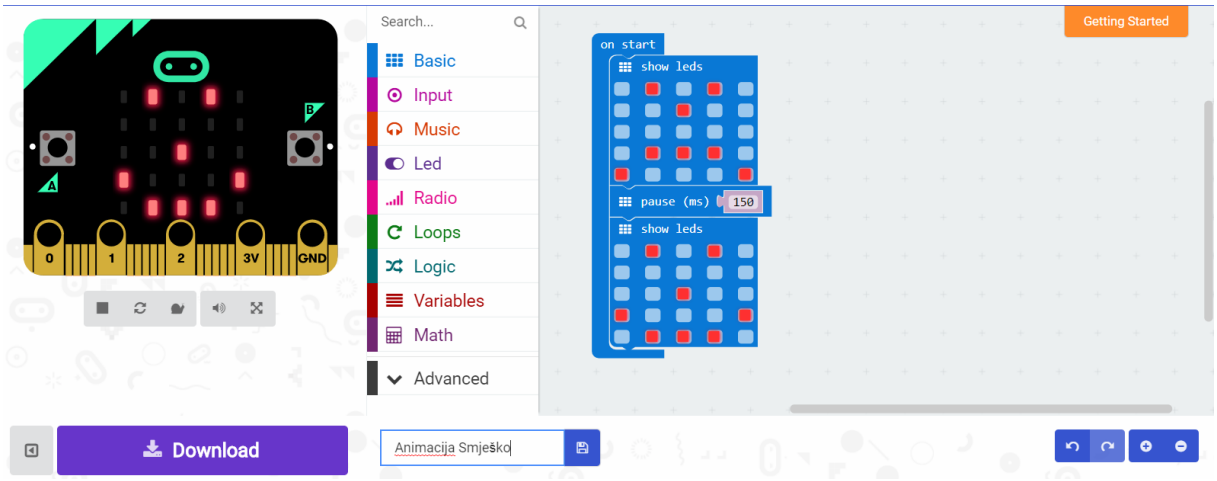
7.1.1. Basic

Prva skupina naredbi se naziva **Basic** skupina. Ona se sastoji od osnovnih naredbi koje su potrebne za funkcioniranje micro:bit programa. Ova skupina se najčešće

koristi tijekom programiranja micro:bita. Unutar skupine nalazi se devet naredbi [15]. Sve naredbe iz ove grupe označene su plavom bojom:

1. **show number** – naredba za prikazivanje brojeva na LED matrici
2. **show leds** – naredba za prikazivanje crteža i slika na LED matrici
3. **show icon** – naredba za prikazivanje već ponuđenih ikona koje se prikazuju na LED matrici (srce, trokut, strelica itd.)
4. **show string** – naredba za upisivanje teksta u program i njegov prikaz na LED matrici
5. **Forever** – beskonačna petlja, pozadinski program se sam pokreće nakon završenog slijeda naredbi
6. **Pause (ms)** – određivanje pauze između naredbi tj. prikaza na LED matrici
7. **On start** – pokretanje programa samo jednom određenom tipkom ili samim uključenjem micro:bit mini-računala
8. **Clear screen** – naredba koja briše cijeli LED zaslon
9. **Show arrow** – prikazivanje strelice, padajući izbornik omogućava određivanje strana svijeta (sjeveroistok, istok, zapad...)

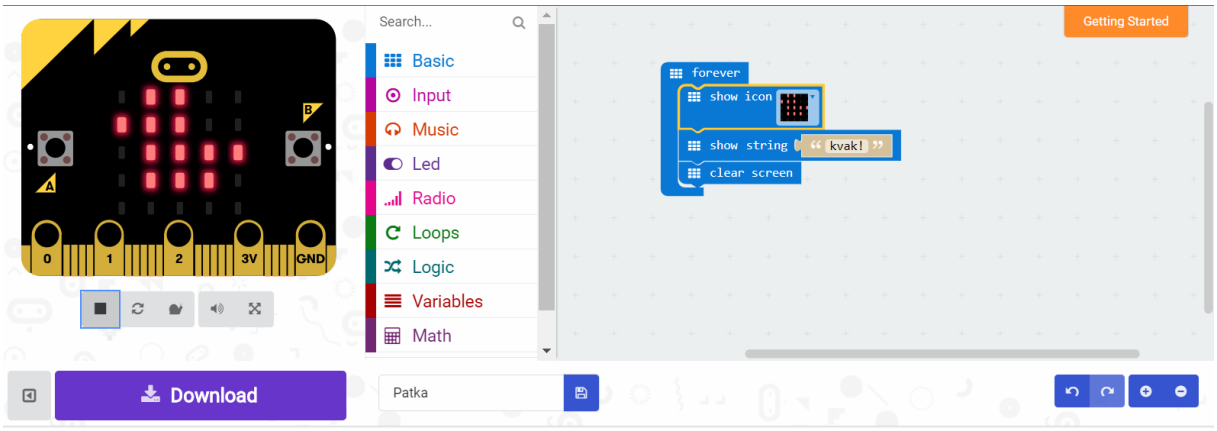
Na primjer, kako bismo prikazali neke od navedenih naredbi u funkciji programa. Možemo osmisliti zadatak koji na simulatoru prikazuje animiranog smješka. Za početak programa biramo hoće li se program pokretati samo jednom ili beskonačni broj puta. Mi smo odabrali naredbu 'on start' koja omogućava provedbu programa samo jednom. Znači, nakon što se sve naredbe učitaju i program se provede, simulator se zaustavlja i stavlja u stanje mirovanja. Naredbom 'show leds' crtamo izgled smješka. Odredit ćemo tužno, te zatim sretno lice. Ovu naredbu ćemo primjeniti dva puta. U prvom dijelu programa sa naredbom 'show leds' crtamo tužno lice, a u drugom dijelu sretno. Između ta dva bloka naredbi ćemo staviti pauzu od 150 milisekundi kako bi prijelaz iz tužnog u sretno lice bio prirodniji. Klikom na gumb 'start the simulator' ispod simulatora, program se izvršava (Slika 24). Prvo prikazuje tužno lice, a zatim sretno. Animacija je spremna za prebacivanje na micro:bit pločicu.



Slika 24. Prikaz programa Animacija smješka

U ovom primjeru programa smo koristili naredbu 'on start' kako bi se blok naredbi proveo samo jednom. Sada ćemo napraviti primjer programa koji se ponavlja u beskonačno. Kako bismo to ostvarili koristimo naredbu 'forever' iz kategorije naredbi Basic. Nakon toga odabiremo naredbu 'show icon' koja se sastoji od mnogo gotovih 'sličica' kao što su srce, majica, nota itd. Mi smo odabrali sličicu patke. Naš program će prikazivati sličicu patke koja je praćena tekstom 'kvak!' kako bismo povezali životinju sa njezinim načinom glasanja. Na kraju programa ćemo zadati naredbu koja briše cijeli LED zaslon te pokreće program ispočetka.

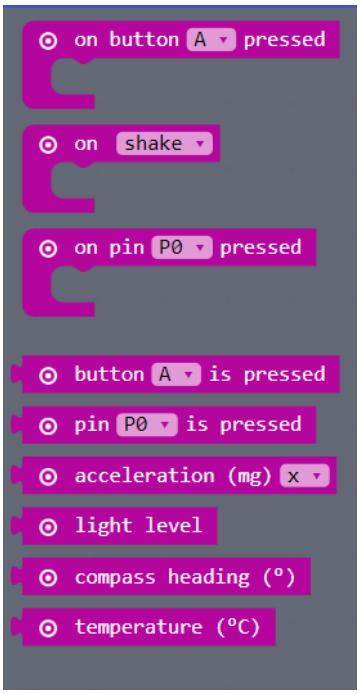
Nakon odabira naredbe 'show icon' slijedi naredba 'show string' u koju se upisuje željeni tekst. Upisali smo riječ 'kvak!'. Naredbom 'clear screen' završavamo blok naredbi programa. Pritiskom na gumb 'start the simulator' pokrećemo program koji se ponavlja dokle god ga ne isključimo (slika 25). Program spremamo i prebacujemo na micro:bit. Nakon završetka prebacivanja programa, isti efekt se vidi na micro:bitu.



Slika 25. Izgled programa 'Patka'

7.1.2. Input

Ova kategorija naredbi koristi ulaze sa senzora koji se nalaze na micro:bitu. Naredbama iz ove kategorije možemo fizički pokretati snimljene programe nakon što ih prebacimo na pločicu. Sastoji se od 14 naredbi koje su označene roznom bojom [16]. Njih koristimo ukoliko želimo koristiti tipke na micro:bit mini računalu (Slika 26).



Slika 26. Prikaz blokova naredbi kategorije 'Input'

1. **On button A pressed** – određena radnja provedena je nakon pritiska na gumb A, B ili kombinaciju gumbova A i B
2. **On shake** – radnja se događa nakon trešnje micro:bit mini-računala
3. **On pin pressed** – radnja se događa nakon što prstima jedne ruke dotičemo tipku GND, a prstima druge ruke dotičemo pin 0, 1 ili 2 micro:bit mini-računala
4. **Button A is pressed** – radnja se događa ukoliko korisnik drži pritisnut gumb A bez otpuštanja
5. **Pin 0, 1 ili 2 is pressed** – radnja se događa nakon što prstima jedne ruke dotaknemo pin GND, a prstima druge ruke istovremeno dotaknemo pin 0, 1 ili 2 te otpustimo ih unutar 1 sekunde
6. **Acceleration (mjeri se u mili G, odnosno 1/1000 G; G je gravitacija zemljine sile teže)** – mjerenje akceleracije (brzine ubrzavanja micro:bit mini-

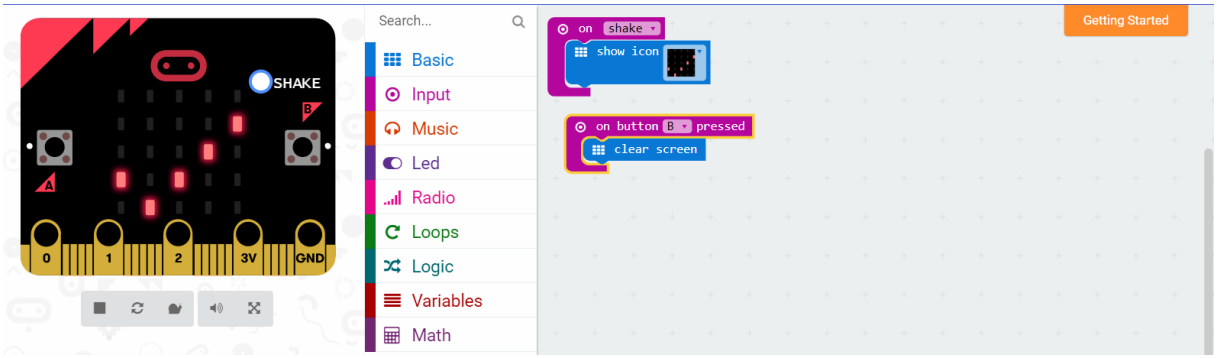
računala) u kojoj X predstavlja smjer okretanja pločice u lijevo ili desno, Y naprijed/nazad, Z gore/dolje i 'strenght' kombinacija X/Y/Z

7. **Light level** – mjerenje svijetlosti koja je uperena prema LED diodama, izražava se u brojevima od 0 do 255, u kojoj 0 predstavlja mrak
8. **Compass heading** – koordinacija prema stranama svijeta koja se računa u stupnjevima, od 0 do 359.
9. **Temperature** – mjerenje temperature senzorom koji se nalazi na micro:bitu, stupnjevima celzijusa
10. **Rotation** – S ovom naredbom može se izmjeriti koliko se micro:bit zarotirao od osnovnog položaja; micro:bit koristi koordinatni sustav sa x i y koordinatama;
11. **Magnetic force** – naredba za mjerenje jačine magnetskog polja. Izmjerena se vrijednost izražava u mikro Teslama
12. **Calibrate compass** – vraćanje kompasa na početni stupanj
13. **Running time (ms)** – naredba koja služi za brojanje vremena izraženo u milisekundama nakon uključenja micro:bita
14. **Running time (micro s**) - naredba koja služi za brojanje vremena izraženo u mikrosekundama nakon paljenja micro:bita
15. **On pin PO released** – naredba koja uključuje 'puštanje' pina PO i GND
16. **Set accelerometer range** – mjerenje brzine ubrzanja micro:bita; najveća brzina ubrzanja može biti do 8 G

Primjer zadatka korištenjem naredbi iz kategorije 'Input', a u kombinaciji s naredbama iz kategorije 'Basic' može se prikazati na mnogo primjera. Biti će prikazana dva primjera. U prvom će se koristiti fizičko potresanje micro:bita, a u drugom tipka „B“. Ovi primjeri će biti spojen u jedan program koji se sastoji od korištenja dva fizički različita način pokretanja programa.

Za početak odabiremo naredbu 'on shake' iz kategorije 'Input'. Ona omogućava izvođenje programa prilikom trešnje pločice. Naredba koju dodajemo unutar tog programa je naredba 'show icon' koja će prikazati sličicu na LED matrici. Uz mnogo ponuđenih sličica mi odabiremo sličicu sa kvačicom. Prvi dio programa je gotov. Nakon što korisnik protrese micro:bit pločicu, na LED matrici se prikazuje sličica kvačice.

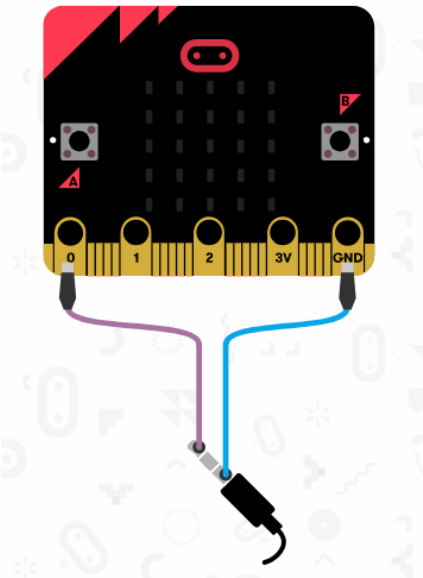
Nakon toga, odabiremo naredbu 'on button B pressed' iz kategorije 'Input' i u njega dodajemo naredbu 'clear screen'. Nakon što korisnik pritisne gumb B, pokreće se naredba koja briše cijeli LED zaslon. U prvom dijelu smo odredili da nakon trešnje micro:bit, korisnik dobiva sličicu kvačice na LED matrici. U ovom dijelu to može ukloniti pritiskom na gumb B (Slika 27).



Slika 27. Izgled programa korištenjem naredbi iz kategorije Input i Basic

7.1.3. Music

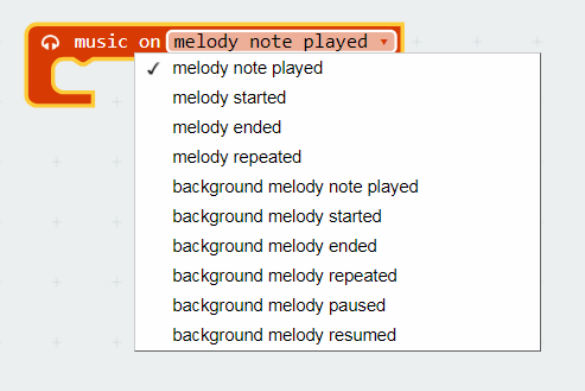
Music je kategorija koja sadrži naredbe koje se odnose na primjenu naredbi u području glazbe na micro:bitu. Naredbe, njih 9, iz ove kategorije označene su crvenom bojom [27]. Za reprodukciju zvuka, potrebno je spojiti klemu na GND, a drugu na oznaku „0“. Kleme je potrebno spojiti sa zvučnikom (Slika 28).



Slika 28. Prikaz pinova „0“ i GND sa spajanjem na kleme koje vode prema zvučniku

U online uređivaču je omogućeno testiranje programa bez potrebe za prebacivanjem programa na micro:bit. Naredbe koje se nalaze unutar kategorije su:

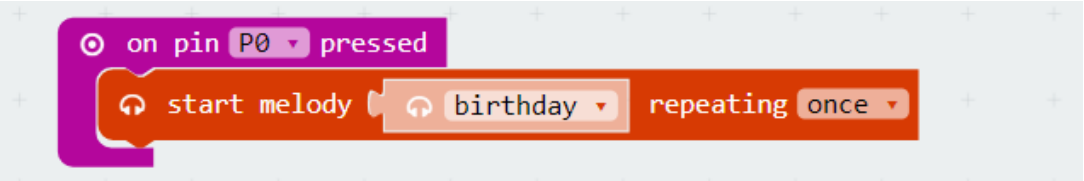
1. **Play tone** – naredba koja pušta zvuk određenog tona, određeni period vremena (engl. beat)
2. **Ring tone (Hz)** – puštanje tonova putem pina PO
3. **Rest (ms)** – glazbena pauza koja se određuje u milisekundama, može se koristiti između tonova kako bi se promijenio ugođaj melodije koju puštamo
4. **Start melody** – započinjanje isprogramirane melodije putem micro:bit pločice i njezinih aktivacijskih gumbova (gumb A, B, pin PO)
5. **Music on** – puštanje kodova za glazbu nakon određene naredbe. Kodovi se sastoje od dvije vrste melodije, jedna je glavna odnosno jedina, a druga se određuje sviranjem u pozadini odnosno ima sporednu ulogu (Slika 29).



Slika 29. Primjeri glavne i sporedne melodije

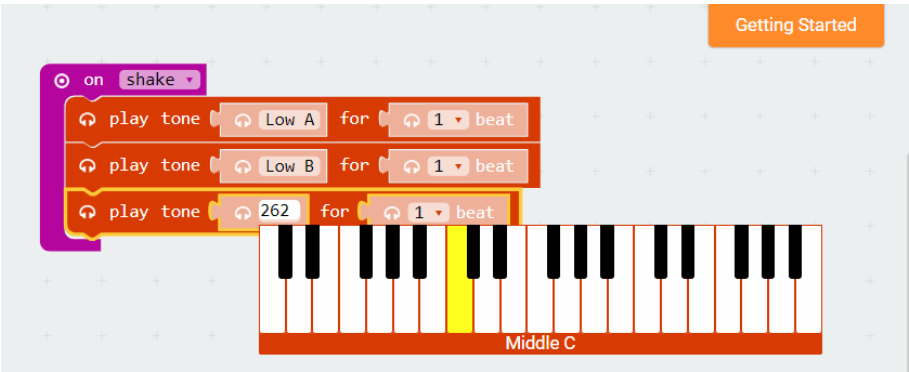
   Nakon pokretanja, zaustavljanja ili sviranja u pozadini određenih melodija koje se mogu ponavljati ili prekidati
6. **Middle C** – tonalitet c-dura, kromatski niz od 12 tonova
7. **Beat** – trajanje melodije izraženo u trajanju doba, otkucaji su izraženi u trajanju jedne dobe, tempom
8. **Tempo (bpm)** –određuje brzinu izvođenja dobe koja se odražava na karakter djela
9. **Change tempo by (bpm)** – mijenjanje brzine izvođenja melodije izraženo u jedinici – beats per minute
10. **Set tempo to (bpm)** – određivanje tempa koji određuje izvođenje glazbenog djela

Kao što smo naveli, korištenje naredbi iz kategorije 'Music' odnosi se na stvaranje i reprodukciju glazbe putem micro:bita. Na primjer, želimo li otpjevati nekom iz razreda rođendansku pjesmicu? To je moguće isprogramirati u micro:bitu. Za početak, odabiremo na koji način želimo pokrenuti sviranje skladbe. To ćemo odabrati naredbom 'On pin PO pressed' iz kategorije 'Input'. Dakle, pritiskom na pin PO program se pokreće. Naredbom 'start melody' iz kategorije 'Music' omogućeno nam je biranje između nekoliko gotovih skladbi. Mi smo odabrali skladbu 'birthday' iz padajućeg izbornika. Nakon toga, možemo birati koliko puta želimo da se odabrana melodija reproducira. Ponuđene opcije su: jednom, beskonačno, jednom u pozadini i beskonačno u pozadini. Mi smo odabrali da se skladba reproducira samo jednom (Slika 30). Skladbu spremamo na micro:bit pod nazivom 'Rođendanska pjesmica'.



Slika 30. Izgled programa 'Rođendanska pjesmica'

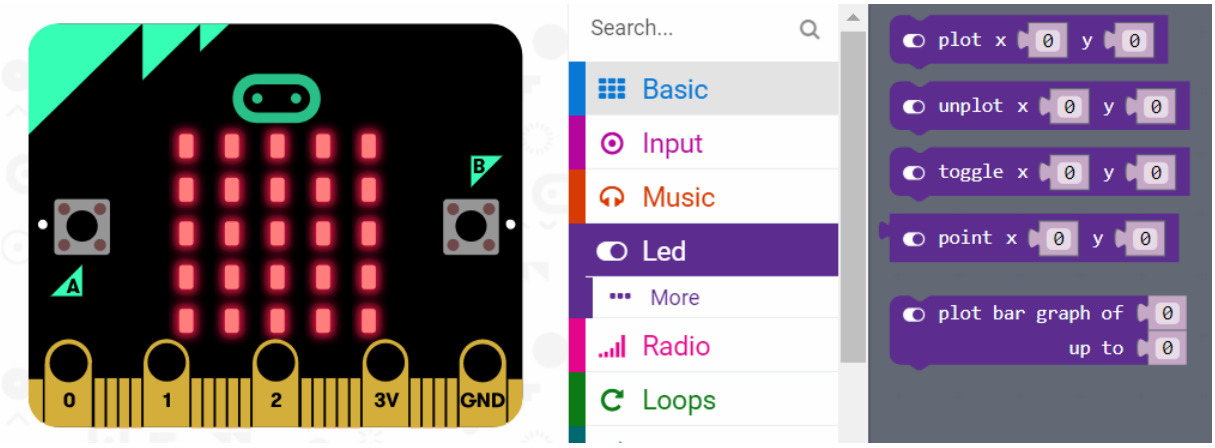
Ukoliko želimo sami stvoriti određenu skladbu. Koristit ćemo se naredbom 'play tone' u kojoj nam je omogućeno biranje tona koji želimo reproducirati putem micro:bit pločice. Za početak određujemo na koji način želimo da se program pokreće. Odabrali smo naredbu 'on shake' iz kategorije 'Input'. Svaki put kada korisnik protrese micro:bit, izvodi se niz naredbi koje smo odredili. Naredbom 'play tone' omogućeno nam je birati tonove koji se reproduciraju. Klikom miša na ime tona u naredbi, otvara se klavijatura na kojoj odabiremo željeni ton. Nakon toga, možemo odrediti trajanje tona tj. **beat.** On može biti u četvrtinkama, osminkama ili cijelim notama. Mi smo odredili trajanje od 1 beat-a (Slika 31). Naredbu 'play tone' koristimo onoliko puta koliko tonova želimo u skladbi. U ovom primjeru smo koristili tri različita tona: Low A, low B i middle C.



Slika 31. Primjer programa koristeći naredbu 'play tone'

7.1.4. Led

Ova kategorija naredbi se odnosi na kontroliranje dioda koje se nalaze na LED matrici micro:bita. Kao što sama riječ govori LED (engl. Light Emitting Diode) diode se odnose na mogućnost prikaza svjetlosnih efekata. Polje dioda sastoji se od 25 dioda [18].



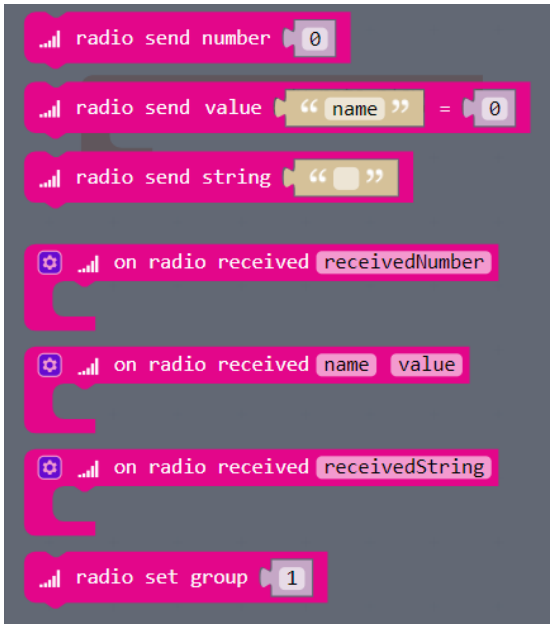
Slika 32. Naredbe 'LED' kategorije

Tako će nam naredbe 'plot' i 'unplot' omogućavati paljenje i gašenje određenih dioda koristeći se X,Y koordinatama. X koordinate pokazuju lijevo-desno, a Y gore-dolje. Naredbom 'toggle' kontroliramo određene diode i možemo napraviti prijelaz iz jednog prema drugom. Što se tiče svjetlosti LED dioda, njih kontroliramo naredbom 'brightness' koja se mjeri u brojčanom obliku od 0 do 255, gdje 0 označava potpuni 'mrak' dok je 255 najjača svjetlost prikaza (Slika 32). Za naprednije programa služi

naredba 'plot bar graph of __ up to' kojom možemo prikazati grafički prikaz određene radnje (Primjer se nalazi u primjeru programa u devetom poglavlju).

7.1.5. Radio

Ova kategorija se odnosi na naredbe za međusobno povezivanje i komunikaciju micro:bitova putem radio valova. Putem emitiranja radio valova (engl. Broadcast) šalju se naredbe i varijable. Na taj način možemo emitirati bilo koji podatak vezan uz radio broj te naziv uređaja u grupi micro:bita na koji se želimo spojiti. Ove naredbe se mogu koristiti za spajanje ili komunikaciju s ostalim micro:bitovima. Isto tako, grupa se može povezati sa zajedničkim indetifikacijskim brojem. Naredbe su označene rozom bojom (Slika 33).



Slika 33. Kategorija naredbi 'Radio'

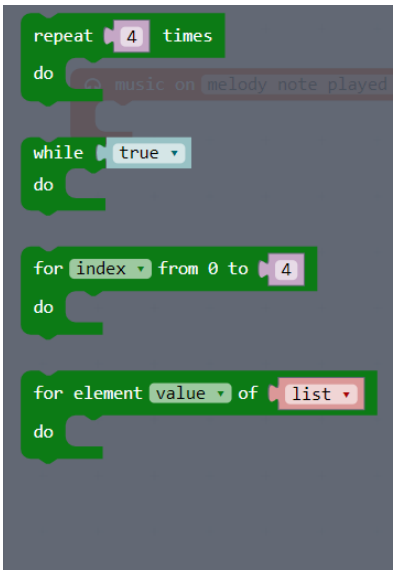
7.1.6. Loops

Jedna od kompleksnijih kategorija bloka naredbi odnosi se na petlje (engl. Loops). One su označene zelenom bojom kako bismo ih lakše prepoznali (Slika 34). Kategorija se sastoji od 4 vrste petlji. Najjednostavnija petlja koja se koristi je petlja ponavljanja odnosno 'repeat-do petlja'. Petlja, 'while-do' se koristi kada želimo da se

petlja ponavlja dok je ispunjen uvjet zadan u petlji. Kad uvjet prestane biti ispunjan, petlja se više ne ponavlja.

Kod treće vrste petlji postoji vrijednosti i argument. Tako, na primjer, 'For index from 0 to 4' funkcionira na način da se vrijednost indeksa postavi na određenu vrijednost (u ovom slučaju na 0) te svakim prolaskom kroz petlju raste za +1. Kad vrijednost dođe do određene vrijednosti (u ovom slučaju do 4) petlja se više ne ponavlja.

'For element value of list petlja' sadrži vrijednost varijable koja se ispunjava zadanim vrijednostima. Svaki atribut koristi zadane vrijednosti i ispunjava slijed naredbi koje je korisnik zadao.



Slika 34. Kategorija naredbi Loops

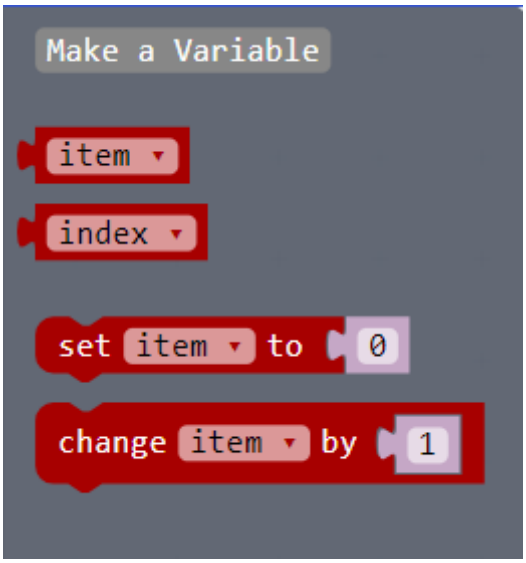
7.1.7. Logic

Ova kategorija naredbi sadrži logički uvjet. Ukoliko je on ispunjen, blok naredbi unutar logičkog uvjeta će se provesti.

Ako je u logičkom uvjetu 'if then' izraz istinit, tada će se blok naredbi izvršiti. Ukoliko izraz nije istinit, blok naredbi će se preskočiti.

7.1.8. Variables

Kategorija naredbi koje sadrže varijable čine temelj svakog programa (Slika 35). One se najčešće nadovezuju na naredbama iz kategorije 'Basic'. Varijable se mogu prikazati u tekstualnom i blokovskom obliku. Isto tako, moguće je postaviti vlastitu varijablu sa nazivom i vrijednošću. Kod programiranja, važno je varijablama dati dobro ime. Svako ime treba biti povezano s namjenom programa ili onim što se programira. Naziv varijable može biti niz od nekoliko slova.



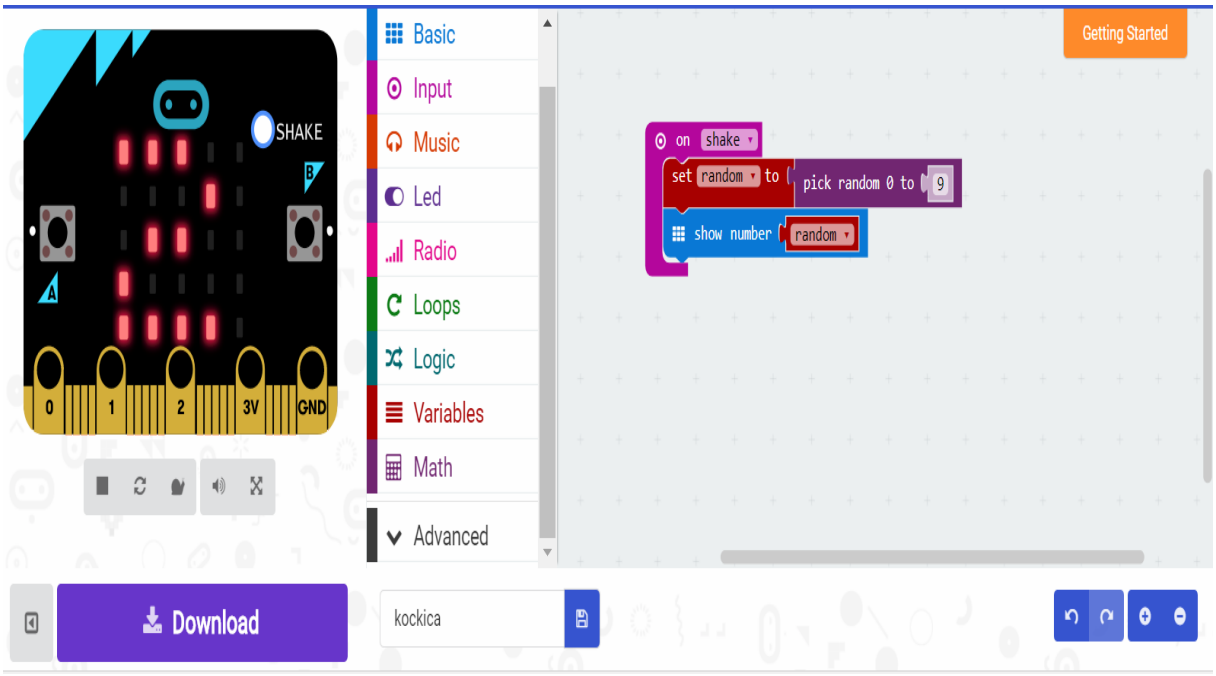
Slika 35. Kategorija naredbi Variables

7.1.9. Math

Aritmetičke operacije koje su ponuđene u ovoj kategoriji bloka naredbi su naredbe za zbrajanje (+), oduzimanje (-), množenje (*) i dijeljenje (/). Blok naredbi se sastoji od dva broja i znakom jednakosti koji nakon provedenog slijeda naredbi određuje rezultat provedbe programa. Korisniku je, također, omogućeno postavljanje uvjeta nasumičnog odabira određene vrijednosti naredbom 'pick random 0 to __' (Slika 36).

Slika 36. Kategorija naredbi Math

Kombinacijom svih naredbi iz ponuđenih kategorija, uključujući i ovu, možemo izraditi kojekakve logičko matematičke zadatke. Na primjer, možemo napisati program koji omogućava korisniku nasumično prikazivanje brojeva od 1 do 9. On funkcionira na način da naredbom 'pick random' programu dajemo samo brojčano ograničenje od kojeg do kojeg broja želimo da on napravi nasumičan odabir. Kako bismo započeli program odabiremo naredbu s kojom ga želimo pokrenuti. Mi smo se odlučili na naredbu 'shake' koja omogućava pokretanje istog trešnjom micro:bit pločice. Iz kategorije naredbi 'variables' odabiremo naredbu 'set item to', toj naredbi dodajemo naredbu 'pick random 0 to' iz kategorije naredbi 'Math' kako bismo odredili raspon brojeva koji će se nasumičnim izborom pojavljivati na LED matrici. Određujemo brojeve od 0 do 9. I za kraj, želimo da se broj pokazuje na LED matrici i time odabiremo naredbu 'show number' iz kategorije naredbi 'Basic'. Program spremamo pod nazivom 'kockica' na micro:bit mini računalo (Slika 37).



Slika 37. Prikaz programa kockica

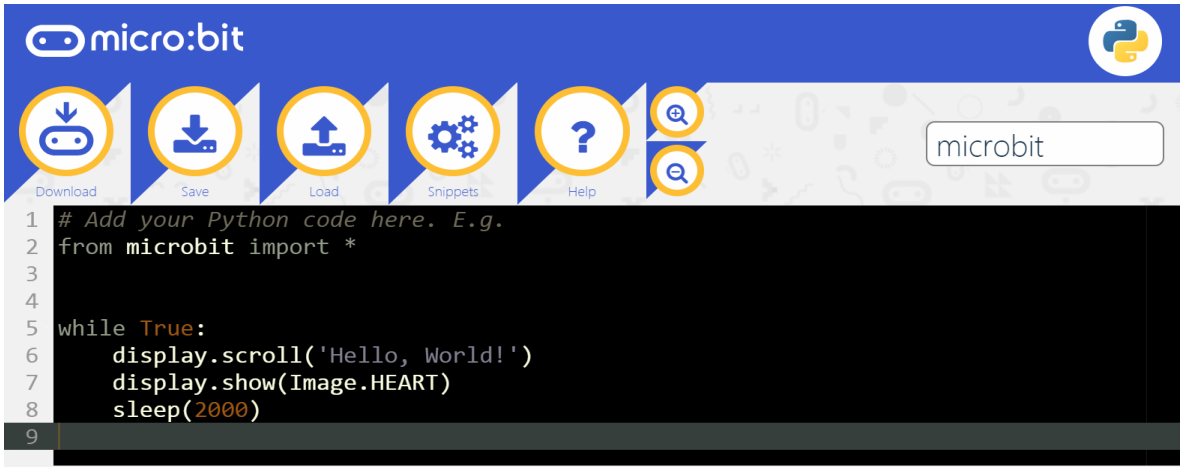
8. Uređivač za pisanje programa u Pythonu

Kao što je ranije navedeno, pisanje programa za micro:bit može biti na osnovnoj – vizualnoj-blokovskoj razini i na naprednoj, tekstualnoj razini. Nakon određenog vremena, kod svakog korisnika se može vidjeti koliko je usvojio programiranje. Ukoliko netko želi unaprijediti svoje vještine, Python je izvrstan način za to. To je programski jezik koji se razvio velikom brzinom. Ponajviše se koristi u razvijanju novih aplikacija, a jednostavnost njegove sintakse omogućuje korisnicima da ga brzo savladaju i brzo napreduju u njemu [10].

Ovaj programski jezik je isključivo tekstualni. Verzija kojom se koristi BBC micro:bit se naziva MicroPython. Ona omogućuje rad na 'malom računalu' kao što je micro:bit [11]. Ne postoji razlika u pisanju kodova za Python i za MicroPython. Određene se naredbe u MicroPythonu ne koriste.

Na web adresi: https://python.microbit.org/v/1 nalaze se besplatni softveri (engl. Software) koji omogućavaju korisnicima korištenje programa i međusobno dijeljenje steknutih iskustva. Cijela micro:bit zajednica se sastoji od volontera koji imaju pravo na povratnu informaciju u vezi programskog koda, programskih grešaka i podršku.

Na slici 38. može se vidjeti kako izgleda sučelje uređivača. Program koji je prikazan koristi se najčešće kao prva vježba prilikom upoznavanja ovog programskog jezika. Naime, program omogućuje korisniku ispis teksta 'Hello, World!' na micro:bit.



Slika 38. Sučelje MicroPython uređivača

Nakon što se ispiše tekst na LED matrici (Slika 39), pojavljuje se i slika u obliku srca. Nakon toga program se stavlja u stanje mirovanja. Dio MicroPythona kôda je:

```
from microbit import *
display.scroll("Hello, World!")
```

Slika 39. Dio programskog koda za program "Hello, World!"

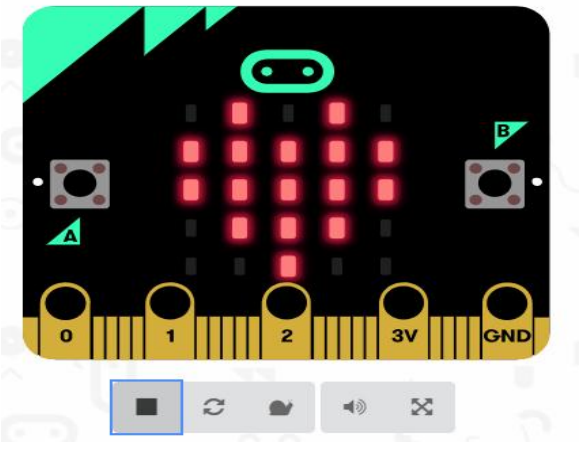
Prva linija koda govori da korisnik želi koristiti podatke o micro:bitu, a oni se nalaze u Python odjeljenju micro:bit modula. Sa znakom 'zvjezdice', dali smo do znanja programu da pretraži cijelo svoje odjeljenje koje je napravljeno za micro:bit (Slika 40). Druga linija koda se odnosi na prikaz tekstualnog dijela "Hello, World!":

```
display.scroll("Hello, World!")
```

Slika 40. Prikaz naredbe za prikaz teksta

Riječ 'display' se odnosi na objekt iz micro:bit odjeljenja (engl. Library) koji predstavlja LED matricu micro:bita. Interpunkcijskim znakom točke, odvajamo dio naredbe u kojoj pozivamo tekst koji će se kasnije prikazati na LED matrici. Naredbom 'scroll' upozoravamo program na tekst koji ćemo unositi u navedene zagrade. Tekst pod navodnim znakovima unutar zagrade određuje argument[1] ovog programskog jezika [12].

Unutar ovog programskog jezika korisnici mogu napisati program kakav žele uz poštivanje sintakse i logičko povezivanje.



Slika 41. Prikaz simbola srca na LED matrici

[1] Argument je vrijednost unutar zagrade, a može biti konstanta ili varijabla.

9. Primjena BBC micro:bita u razrednoj nastavi

Do sada je STEM revolucija uvedena odnosno primijenjena u predmetnoj nastavi osnovne škole, srednjim školama i na određenim visokoobrazovnim ustanovama. U ovom diplomskom radu predložit ću njezinu primjenu u razrednoj nastavi. Na području obrazovanja dolazi do velikih promjena. Primjer tome je kurikularna reforma koja će omogućiti strukturne promjene i uspostavljanje učinkovitog sustava odgoja i obrazovanja. Samim time zalaže se za uvođenje predmeta Informatike od prvog razreda osnovne škole. Uz tu činjenicu, držim kako bi se rad sa micro:bitom mogao uključiti u STEM područje. Djeca će od najmlađe dobi upoznati digitalni svijet i razvijat će logičko mišljenje kroz igru. Algoritamski načini poučavanja putem programiranja u BBC micro:bitu omogućit će rastavljanje programskog zadatka na pojedine dijelove i njihovo spajanje u cjelinu. U nastavku je primjer korištenja micro:bita u pojedinim predmetima u razrednoj nastavi.

9.1. Matematika

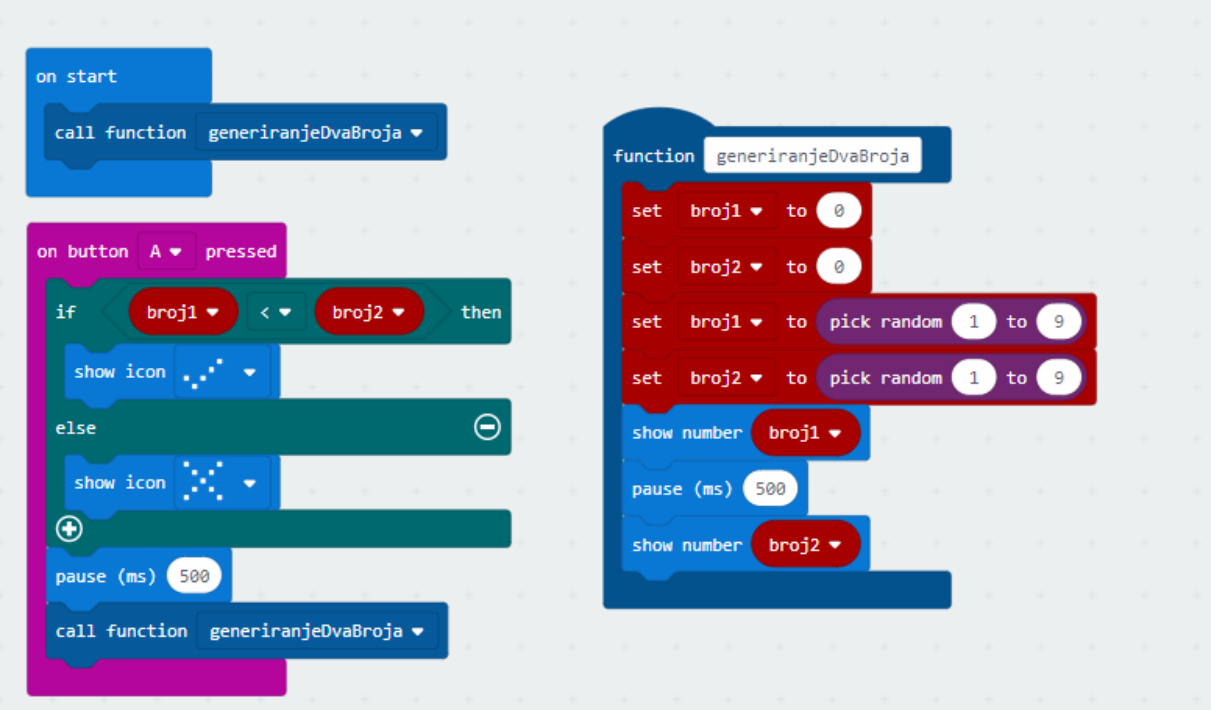
Uspoređivanje brojeva – manji, veći ili jednak

Program u kojem učenik određuje je li prvi broj veći, manji ili jednak drugom. Kad učenik pritisne tipku A – tada je brojka manja, tipka B- veća, a kombinacijom tipki A+B jednaka.

Za početak ćemo odrediti kako će se pokrenuti program. Naredbom 'on start' iz kategorije 'Input' određujemo pozivanje funkcije i početak provedbe naredbi u programu. Funkciju smo odabrali iz kategorije 'Functions' i nazvali je 'Generiranje dva broja'.

U sljedećem koraku određujemo što funkcija radi. Želimo da se nasumičnim odabirom biraju brojevi od 1 do 9. Učenik dobiva dva broja i kasnije se određuje koji od ta dva broja je manji, veći ili jednak. Naredbom 'set number to' iz kategorije 'variables' postavljamo brojeve na nulu. Kako bismo kasnije pozivali te brojeve od 1 do 9 nasumičnim odabirom. Postavili smo početne vrijednosti brojeva. Nakon toga, dodajemo jos dva puta istu naredbu kako bi za svaki broj dodali određeni raspon (od 1 do 9). Naredbama 'set number to' dodajemo naredbu 'Pick random to' i unosimo vrijednosti od 1 do 9. Kako bismo prikazali broj 1 i broj 2 u funkciji. Naredbom 'show

number' određujemo brojčani prikaz istih. Naredbe smo postavili iz kategorije 'Basic'. Između ta dva prikaza brojeva, stavljamo pauzu od 500 milisekundi kako bi učenik stigao pročitati prvi, a zatim drugi nasumično izabran broj sa LED matrice (Slika 42).

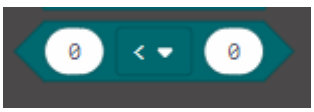


Slika 42. Program Uspoređivanje brojeva – određivanje funkcije

Za biranje određenih funkcija programa, koristit ćemo tipku A, tipku B i kombinaciju tipki AB, te na kraju naredbu 'shake' čije pokretanje se vrši trešnjom micro:bit pločice.

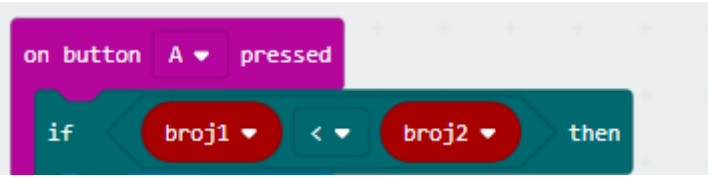
Tipka A

Naredbom 'on button A pressed' iz kategorije 'Input' određujemo funkciju tipke A u programu. Ovdje ćemo korištenjem 'if then else' petlje postaviti uvjet ukoliko je broj 1 manji od broja 2. Program će prikazati točnost uvjeta samo ako je prvi broj veći od drugog. I to na način prikazivanje simbola kvačice ili iksa na LED matrici. Petlju 'if then else' pozivamo iz kategorije Logic. Nakon toga iz iste kategorije odabiremo naredbu 'comparison' kako bismo odredili uvjet petlje (Slika 43). Broj 1 mora biti manji od broja 2.



Slika 43. Naredba 'comparison' iz kategorije Logic

Određujemo znak 'manji od' te na mjesto brojeva 0, dodajemo naredbu 'Broj 1' i 'Broj 2' koje smo postavili u kategoriji Variables.



Slika 44. Prikaz postavljanja uvjeta

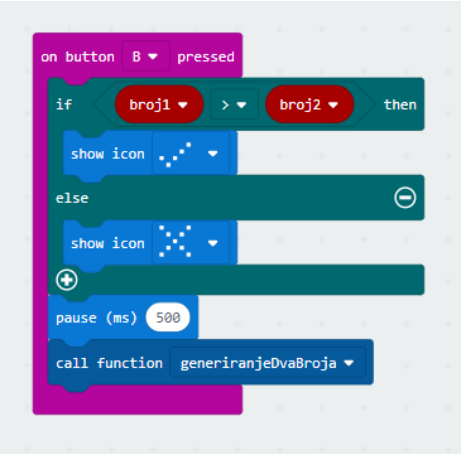
Nakon toga moramo prikazati simbol kvačice ukoliko je uvjet istinit. Naredbom 'show icon' iz kategorije Basic odabiremo simbol kvačice. Naredbu postavljamo ispod uvjeta if prikazanog na slici 44. Ukoliko uvjet nije ispunjen, želimo da program prikaže simbol iksa na LED matrici. Ponovo koristimo naredbu 'show icon' i ovaj put odabiremo simbol iksa te ga postavljamo ispod uvjeta 'else' petlje 'if then else'. Na završetku uvjeta postavljamo pauzu od 500 milisekundi kako bi simbol kvačice ili iksa bio vidljiv učeniku. Na kraju uvjeta ponovno pozivamo funkciju 'Generiranje dva broj' kako bi se mogla dalje koristiti. Naredbom 'call function' iz kategorije Functions pozivamo istu.

Tipka B

U funkciji tipke B želimo da uvjet prikazuje da je broj 1 veći od broja 2. Naredbom 'on button B pressed' iz kategorije Input pozivamo određivanje uvjeta unutar funkcije tipke B. Koristimo isti uvjet kao sa tipkom A.

Kako ne bismo sve radili ispočetka kopirat ćemo program koji smo napisali za funkciju tipke A, desnim klikom miša kliknemo na blok naredbi 'on button A pressed' i odaberemo naredbu 'duplicate'.

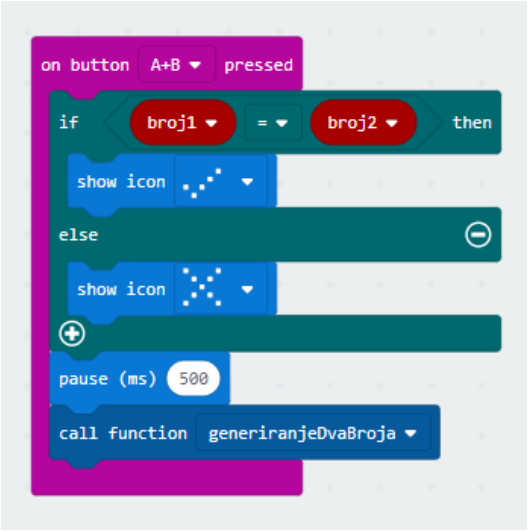
Povlačenjem duplikata odvlačimo ga na blok naredbi 'on button B pressed' te izmijenimo željeni uvjet. Odabiremo znak 'veći od' u uvjetu if. Između brojeva 1 i broja 2. Ostale naredbe ostaju iste kao u prijašnjem bloku naredbi (Slika 45).



Slika 45. Blok naredbi funkcije tipke B

Kombinacija tipki AB

Kombinacijom tipki A i B želimo prikazati uvjet u kojem je broj 1 jednak broju 2. Na isti način određujemo dupliciranje uvjeta kao u prethodnom koraku. Desnim klikom miša određujemo dupliciranje te ga povlačimo na prazan prostor za stvaranje kodova. Naziv naredbe 'on button B pressed' mijenjamo u 'on button AB pressed'. To određujemo klikom miša na padajući izbornik pokraj slova B. Između varijable 'broja 1' i varijable 'broja 2' odabiremo lijevom tipkom miša padajući izbornik i biramo opciju '='. Postavili smo uvjet za kombinaciju tipki AB ukoliko je broj 1 jednak broju 2.



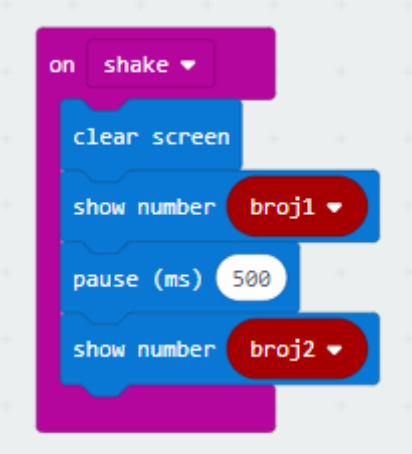
Slika 46. Dio programa za kombinaciju tipki AB

Naredba 'on shake'

U ovom bloku naredbi odredit ćemo nasumičan odabir broja 1 i broja 2 navedenog programa. Cilj programa je trešnjom micro:bit pločice odrediti nasumičnim odabirom broj 1 i broj 2. Iz kategorije Input odabireo naredbu 'on shake' i postavljamo ju na prazan prostor za kodiranje.

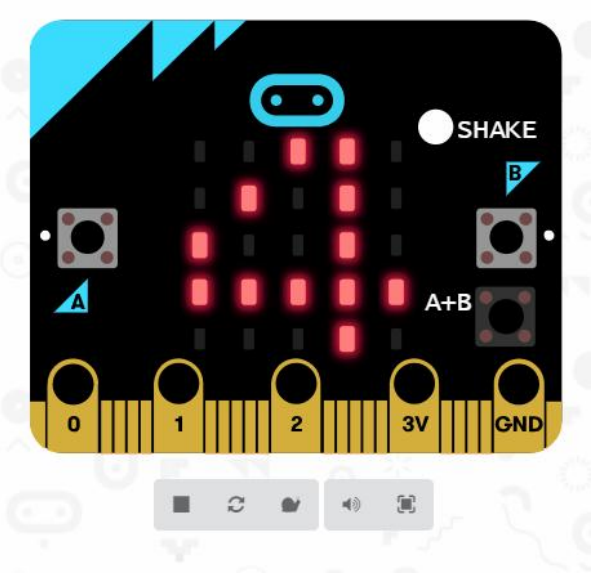
Nakon toga određujemo naredbom 'clear screen' iz kategorije Basic kako bi program nakon trešnje očistio ekran i tek onda prikazao prvi zadani broj. Naredbom 'show number' određujemo prikaz prvog broja.

U područje te naredbe dodajemo varijablu 'Broj 1' kako bi program prvo nasumično birao 'Broj 1', a zatim 'Broj 2' zadane funkcije. Istim načinom pozivamo i prikaz broja 2 te umetamo varijablu s nazivom 'broj 2'. Između biranja broja 1 i broja 2, potrebno je staviti pauzu od 500 milisekundi kako bi učenik stigao pročitati koji brojevi su zadani.



Slika 47. Dio programa određen trešnjom micro:bit-a

Program je spreman za testiranje. Na simulatoru možemo testirati program ili ga prebaciti na micro:bit. Kombinacijom tipki provjeravamo ispravnost postavljenih uvjeta.



Slika 48. Rezultat programa za uspoređivanje brojeva

Određivanje parnog i neparnog broja

Cilj ovog programa je prikazivanje parnih i neparnih brojeva. Nasumičnim odabirom program vrti brojeve od 0 do 9. Ukoliko je broj paran, na LED matrici se prikazuje simbol kvačice. Ukoliko je broj neparan, na LED matrici se prikazuje simbol iksa.

Za početak određivamo pozivanje funkcije. Iz kategorije Functions odabiremo naredbu 'call function' i dajemo joj ime 'Random broj'. Uz pomoć varijabli ćemo postaviti vrijednosti varijable za odabiranje nasumičnih brojeva od 0 do 9.

Naredbom 'set item to 0' postavljamo varijablu na 0. Ime 'item' preimenuje se na 'Broj' kako bismo znali da je to broj koji će program nasumično prikazati na LED matrici. Dupliciramo zadanu naredbu i na mjesto nule dodajemo naredbu 'pick random 0 to 9' iz kategorije Math. Postavljamo raspon brojeva od 0 do 9 (Slika 49). Naredbom 'show number' prikazujemo njegov prikaz na LED matrici.

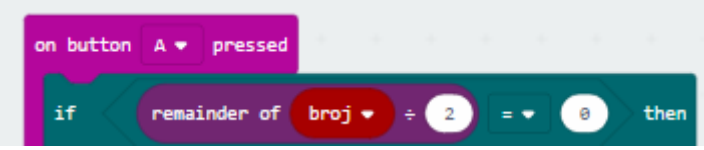
Slika 49. Određivanje blokova naredbi funkcije Random broj

Nakon zadavanja funkcije moramo odrediti na koji način se provodi program. Funkcija bira nasumično brojeve od 0 do 9, a cilj programa je prepoznati parne i neparne brojeve. Zbog toga ćemo odrediti funkcije tipke A koja će prikazivati parne brojeve i funkciju tipke B koja će prikazivati neparne brojeve.

Tipka A

Tipkom A ćemo odrediti prikaz parnih brojeva. Kako bismo odredili samo parne brojeve, postavit ćemo uvjet koristeći petlju 'if then else' iz kategorije Logic. Za početak ćemo naredbom 'on button A pressed' iz kategorije Input odrediti funkciju programa pritiskom na tipku A.

Nakon toga smo dodali petlju 'if then else'. Naredbom 'comparison' iz kategorije Logic dodajemo uvjetu If određivanje parnih brojeva. Na mjesto prvog broja ćemo dodati naredbu 'reimander of 0/1'. U njega ćemo ubaciti varijablu 'Broj' koju smo dodali u kategoriju Variables. Nakon toga postavljamo uvjet u kojem program bilo koji nasumično odabrani broj dijeli s brojem 2 (kojeg smo upisali u naredbu 'remainder of..). Ukoliko je broj dijeljiv s brojem 2, broj je paran te je jednak (==) 0 odnosno nema ostatak i na LED matrici se prikazuje simbol kvačice (Slika 50).



Slika 50. Dodavanje varijable 'Broj'

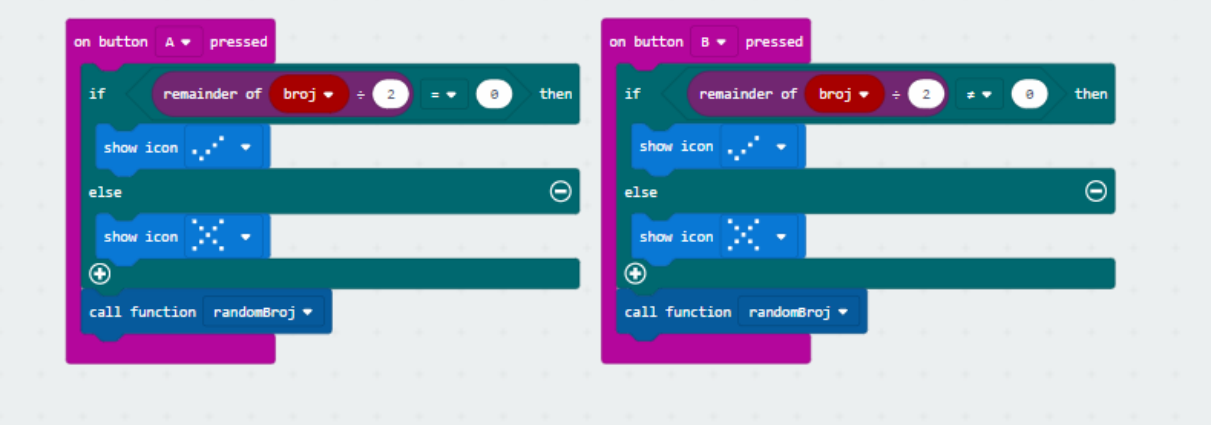
Nakon toga ispod uvjeta if, dodajemo simbol kvačice. Naredbom 'show icon' iz kategorije Basic odabiremo željeni simbol i dodajemo ga u program. Ispod uvjeta else na isti način dodajemo simbol kvačice. U posljednjem dijelu trenutnog bloka naredbi, ponovno pozivamo funkciju 'Random broj' kako bi se uvjet mogao ponovo izvesti i odabrati neki nasumičan broj.

Na ovaj način smo postavili uvjet u kojem program prikazuje parne brojeve. Na isti način ćemo postaviti blok naredbi s aktivacijskom tipkom B. Zbog toga ćemo kliknuti na trenutni blok naredbi i kopirati ga na prazan prostor za kodiranje.

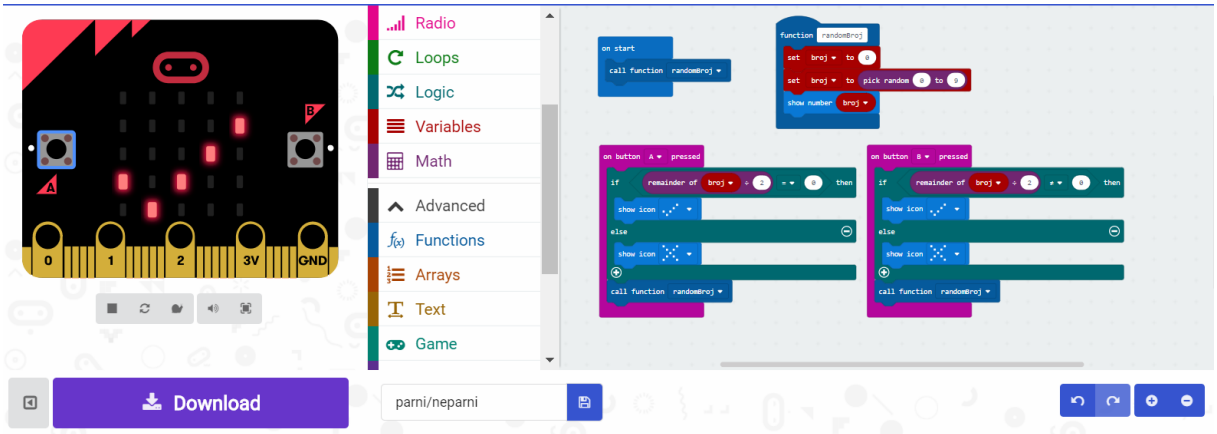
Nakon toga ćemo u padajućem izborniku naredbe 'on button A pressed' promijeniti na slovo B. U if uvjetu ćemo promijeniti znak '=' u znak nejednakosti '≠'. Na taj način smo postavili drugi uvjet koji određuje neparne brojeve. Dijeljenjem nasumično

odabranog broja, program rezultat prikazuje s 'ostatkom' odnosno nejednakošću podijeljenog broja. Samim time možemo zaključiti da je broj neparan (Slika 51).

Naredbom 'on start' određujemo pozivanje funkcije Random broj kako bismo pokrenuli stvoreni redoslijed blokova naredbi (Slika 52).



Slika 51. Program za određivanje parnih i neparnih brojeva



Slika 52. Izgled programa parni/neparni brojevi

Brojevna crta

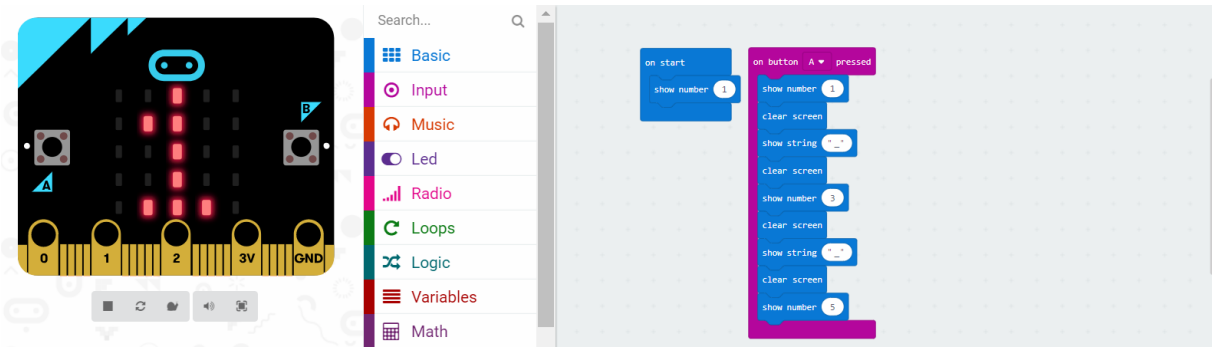
Zadatak učenika je upisati u bilježnicu brojeve koji nedostaju. Program prikazuje niz od 5 brojeva. Svaki drugi broj u nizu nedostaje (1,3,5). Zadatak učenika je pokrenuti program i odgonetnuti koji broj sa brojevne crte nedostaje. Brojevi koji nedostaju, taj dan se koriste za vježbanje i ponavljanje zbrajanja.

Program započinje naredbom 'on start' kategorije Basic. Time smo odredili da će se program pokrenuti samo jednom. Naredbom 'show number' iz iste kategorije

određujemo ispis broja 1. Samim time određujemo početak zadatka. Učenik će vidjeti preostali niz tek nakon što pritisne tipku A na micro:bit pločici.

Naredbom 'on button A pressed' određujemo blok naredbi za ostatak zadatka. Iz kategorije Basic odabiremo naredbu 'show number' te upisujemo broj 1. Kako bismo vizualno lakše raspoznali što se pojavljuje na LED matrici, poslije svake naredbe unosa teksta i brojeva, odredit ćemo naredbom 'clear screen' brisanje LED matrice kako bi se brojevi i tekst pojavljivali jedan po jedan, a ne u nizu. Iz kategorije Basic odabiremo naredbu 'clear screen'. Naredbom 'show string' unosimo znak '_' koji označava rupu na brojevnoj crti. Zadatak učenika je prepoznati ono što nedostaje. Nakon toga ponovno ubacujemo naredbu 'clear screen' te nakon nje 'show number 3'.

Zasad smo unesli upis brojeva '1_2_3'. Dodavanjem naredbe 'show string' upisujemo znak '_' i ubacujemo 'clear screen'. To ponavljamo i za unos broja 5 i znaka '_'(Slika 53).



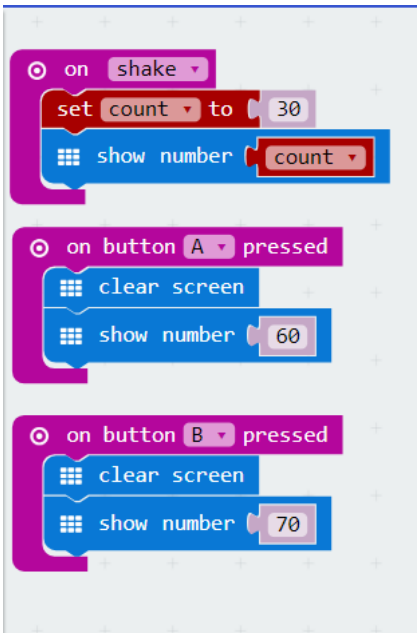
Slika 53. Izgled zadatka 'Brojevna crta'

Čitanje i pisanje desetica

Ovaj zadatak je napravljen od tri različite naredbe. Naredbom 'shake' omogućujemo LED matrici da promjenom smjera trešnje pločice, pokrećemo program koji prikazuje broj 30.

Zadatak učenika je pročitati broj na glas i zapisati njegov naziv riječima na ploču. Nakon toga slijedi drugi dio zadatka koji smo prikazali pritiskom na gumb A. U tom trenutku se program zaustavlja i prikazuje se novi broj, broj 60. Nakon klika na gumb B, prikazuje se novi broj (70) koji je potrebno napisati riječima kao i predhodna dva.

Na ovaj način smo objedinili korištenje naredbe 'shake' i aktivacijskih gumbova A i B (Slika 54).



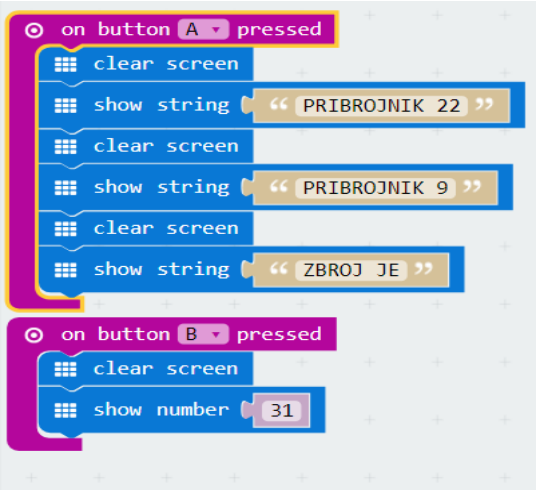
Slika 54. Prikaz programa 'Čitanje i pisanje desetica'

Zbrajanje

Program koji prikazuje brojevni niz i postavlja zadatak u kojem učenik mora prepoznati o kojoj se operacijskoj radnji radi kako bih ga uopće mogao riješiti. Koristimo naredbe 'on button A pressed' i 'on button B pressed' kako bismo unosili tekst postavljenog zadatka.

U prvom dijelu zadatka smo unijeli tekst 'pribrojnik 22, pribrojnik 9' te 'zbroj je' kako bi učenik mogao postaviti zadatak i izračunati ga. Pritiskom na gumb B provjerava rješenje postavljenog zadatka, a to je broj '31'.

Naredbom 'show string' unosimo tekstualni i brojčani izraz. Naredba 'clear screen' spriječava preklapanje unesenog teksta i omogućava postupno prikazivanje zadatka (Slika 55).

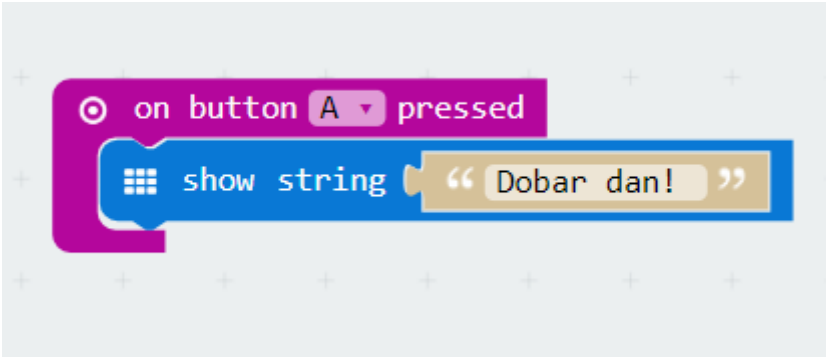


Slika 55. Prikaz zadatka 'Zbrajanje'

9.2. Hrvatski jezik

Upisivanje teksta

Zadatak ovog programa je ispisati vlastito ime na micro:bit. Naredbom 'show string' je omogućen upis teksta. Kako bi program mogao pokrenuti ispis moramo odrediti na koji način ćemo pozvati program. Naredbom 'on button A pressed' program se pokreće u obliku teksta na LED matrici (Slika 56).



Slika 56. Prikaz blokova naredbi za program 'Upisivanje teksta'

9.3. Priroda i društvo

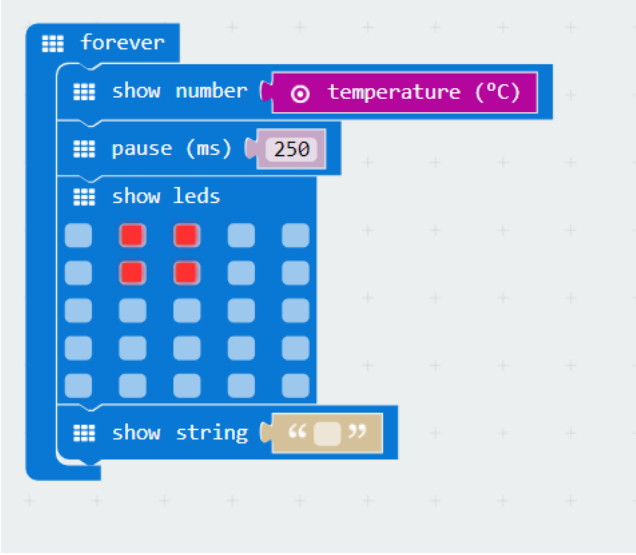
Termometar

U ovom zadatku ćemo objasniti kako je moguće napisati program koji će prikazati temperaturu zraka u prostoriji. Cilj programa je korištenjem LED matrice, prikazivanje temperature i njezina trenutna vrijednost (Slika 57).

Kako bismo došli do toga, koristit ćemo petlju 'forever'. Na taj se način program sam pokreće i traje dok se ne promijeni program ili micro:bit ugasi. Program smo podesili na sljedeći način:

Iz kategorije 'Basic' smo odabrali naredbu 'show number' te umjesto prikazivanja brojčane vrijednosti, umetnuli smo naredbu 'temperature' iz kategorije 'input'. Zbog ovih naredbi, micro:bit koristi senzore temperature i omogućava programu trenutno mjerenje iste. Nakon prikaza brojčane vrijednosti temperature, odredili smo vremensku pauzu zadržavanja brojke na LED matrici od 250 milisekundi.

To smo učinili naredbom 'pause (ms)' iz kategorije 'Basic'. Nakon pauze smo prikazali simbol koji predstavlja stupnjeve celzijusa, koristeći naredbu 'show leds'. Klikom miša na njezine 'diode' označili smo oblik simbola. Na kraju programa je postavljen blok naredbi koji prikazuje prazno polje teksta. Na taj način se neće ispisati vrijednost temperature u trenutku kada se prikazuje simbol stupnjeva Celsijusa (°C).



Slika 57. Prikaz blokova naredbi za Termometar

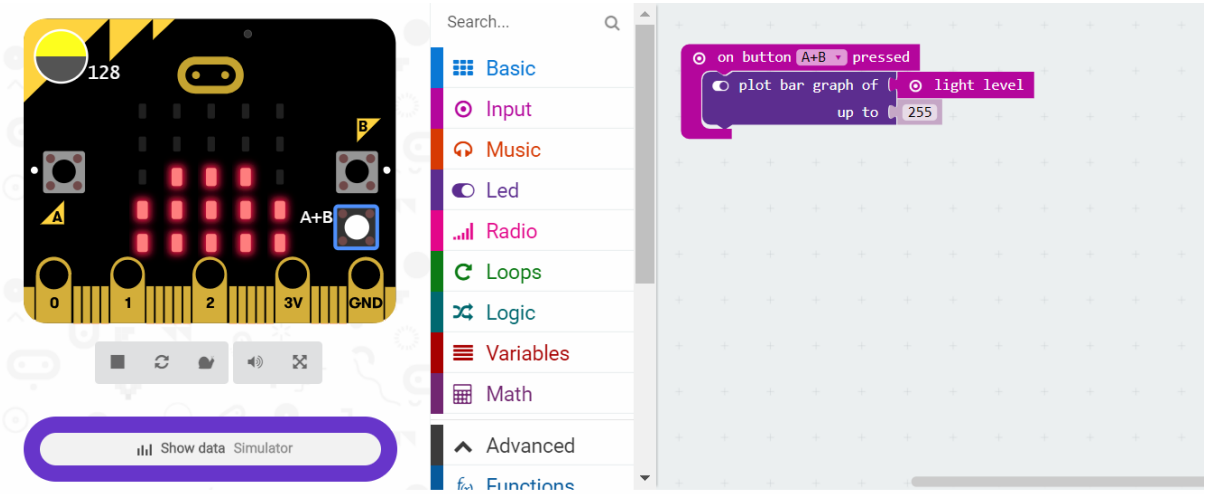
Mjerač svjetlosti

Program koji koristi naredbe za prikazivanje grafičkih vrijednosti na micro:bitu. Nakon što se program izvede, korisniku je omogućeno mjerenje svjetlosti prostorije u kojoj se nalazi. LED diode i senzor koji se nalazi na pločici, omogućavaju grafički prikaz jačine svjetlosti.

Program se pokreće pritiskom na kombinaciju tipki A + B, koje određujemo naredbom 'on button pressed' iz kategorije naredbi 'Input'.

Za početak odabiremo beskonačnu petlju iz kategorije naredbi 'Basic' koristeći naredbu 'forever'. Kao što je prije napomenuto, na ovaj način se program izvodi bez stajanja. Iz LED kategorije naredbi odabiremo naredbu 'plot bar graph' koja omogućava grafički prikaz određene veličine na ekranu micro:bita. Kako bismo izmjerili količinu svjetlosti koristimo se naredbom 'light level' iz kategorije 'Input'.

Uz 'up to' argument, postavlja se najveća vrijednost koju micro:bit može prikazati, a to je brojčano 255 (Slika 58) .



Slika 58. Prikaz blokova naredbi za program 'Mjerač svjetlosti' i prikaz na simulatoru

Korisnik može vidjeti konkretne vrijednosti nakon što prebaci program na micro:bit. Promjenu temperature je moguće isprovocirati držanjem micro:bita u ruci ili približavanjem nekom izvoru topline.

Kompas

Jedna od osnovnih stvari koja se uči u razrednoj nastavi je orijentacija u prostoru. Micro:bit nam omogućava prilagodbu programskog koda kako bi se mogli orijentirati putem kompasa [14].

Unutar prostora za pisanje koda postavljamo beskonačnu petlju 'forever' koja označava trajanje izvedbe programa. Iz kategorije naredbi 'Logic' odabiremo petlju 'true or folse' koja označava istinitost prvog uvjeta. Ukoliko je prvi dio bloka naredbi točan, program se izvodi, ukoliko nije, izvodi se drugi dio petlje. U petlju dodajemo naredbu 'an integer number' iz kategorije 'Logic' koja govori o jednakosti dvije vrijednosti. Ukoliko su one istinite, vraćamo se na 0. Zatim, iz kategorije 'input' dodajemo naredbu 'compass heading' u prvi dio petlje.

S tom petljom ćemo odrediti uvjete u stupnjevima koji će pokazivati određene strane svijeta na micro:bit kompasu. Nakon toga dodajemo argument u petlju. Argumente dodajemo na način da pritisnemo plavi aktivacijski gumbić na 'if then else' petlji Metodom 'drag and drop' odabiremo 'else if' naredbe kako bismo napisali programski kod za sve 4 strane svijeta koje će kompas prikazivati.

Naredbom 'an integer number' određujemo od koliko do koliko stupnjeva se nalazi Istok, Zapad, Sjever ili Jug. Razmak u stupnjevima će biti 90 stupnjeva po strani svijeta (360/4=90).

Nakon toga, dodajemo naredbu 'show string' iz kategorije naredbi 'Basic' kako bismo upisali ime strane svijeta na koju će se ti stupnjevi odnositi. (Slika 58)

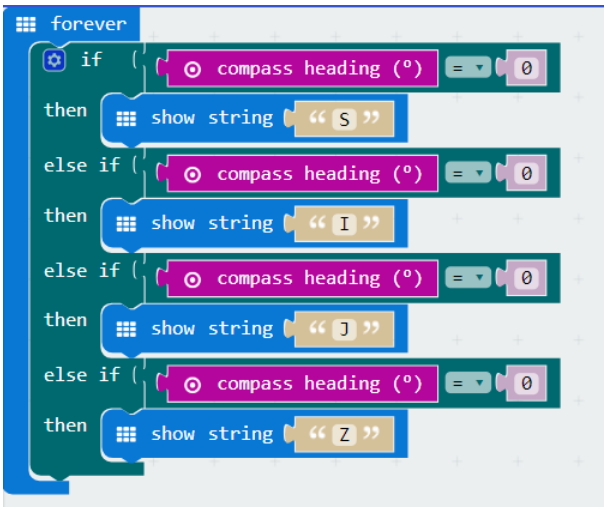
Argumente u petnji ponavljamo 4 puta. Svaki blok naredbi se odnosi na jednu stranu svijeta (Slika 60):

S – Sjever

Z – Zapad

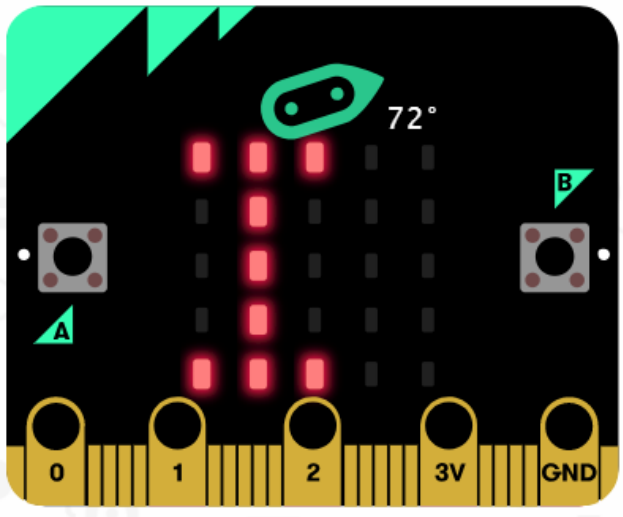
I – Istok

J - Jug



Slika 58. Upisivanje strana svijeta

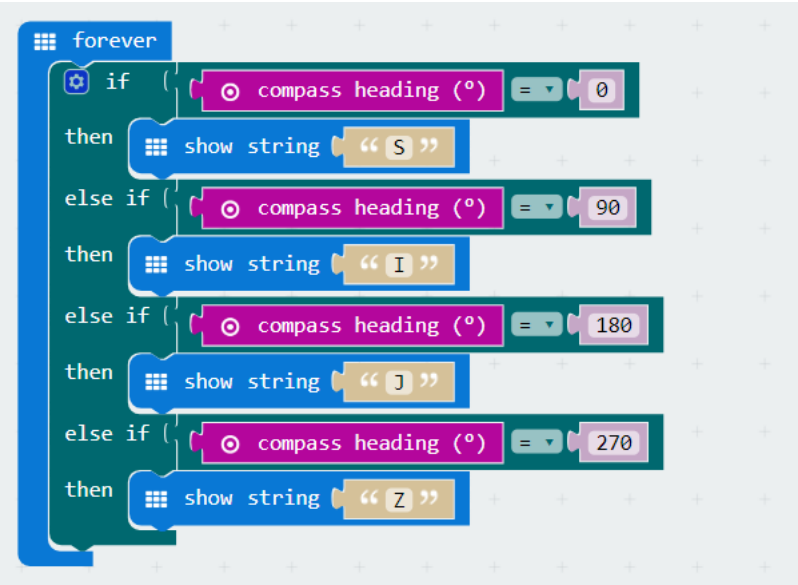
Kompas broji stupnjeve od desne strane u lijevu, što smješta Sjever na 0°C, stoga upisujemo u prvu brojčanu vrijednost 0, kako bi se za istinitost te tvrdnje pokazao 'S' Sjever na LED matrici. S desne strane se nalazi Istok, što znači da moramo zakrenuti kompas za 90°C u desno. U naredbi za upisivanje teksta upisujemo znak 'I' što označava Istok. Jug se nalazi na 180°C, a Zapad na 270°C. Na simulatoru možemo provjeriti kako to izgleda kada vrtimo micro:bit u određenu stranu (Slika 59).



Slika 59. Simulator programa Kompasa

U gornjem dijelu se pojavljuje pomični dio simulatora na koji kada kliknemo lijevom tipkom miša možemo rotirati kompas u bilo kojem smjeru. Brojčano se prikazuje stupanj zakrenutosti, a naredbe iz petlje prikazuju o kojoj strani svijeta se radi. Na

primjeru slike možemo vidjeti da se kompas zakrenuo za 72°C u desnu stranu i time pokazuje na Istok.



Slika 60. Programski kod za program Kompas

9.4. Likovna kultura

Kontrast svijetlo – tamno

Micro:bit omogućava promjenu jačine svijetlosti koristeći LED diode. Njezine diode ćemo koristiti kao senzor. Program se sastoji od dva dijela – prvi dio je zaslužan za prikazivanje oblika na LED matrici i koristi naredbu 'on start', a drugi dio programa koristi beskonačnu petlju 'forever' koja omogućava varijabli 'change item' postupno mijenjanje jačine svijetlosti postavljenog efekta (Slika 61).

U prvom koraku odabiremo naredbe 'on start' i 'forever' iz kategorija naredbi 'Basic'. Kao što smo objasnili, programski kod će se sastojati od dva dijela.

Prvi dio programa:

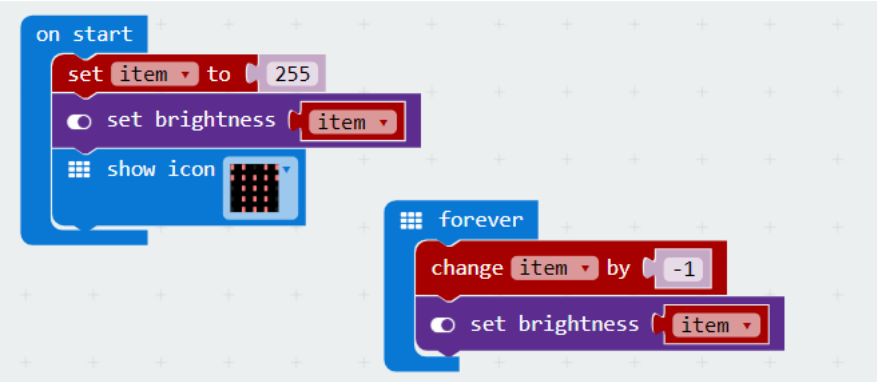
Uzimamo naredbu iz kategorije 'Variables' pod nazivom 'set item to'. Iz LED kategorije naredbi odabiremo 'set brightness' s brojčanom vrijednošću jer svaka boja ima svoj brojčani zapis. Mi ćemo vrijednost jačine svijetlosti (engl. brightness)

podesiti na 255. Kako bismo odredili oblik koji želimo da mijenja jačinu svijetlosti, iz kategorije naredbi 'Basic' koristimo naredbu 'show icon'. Odabrali smo simbol majice (engl. t-shirt).

Drugi dio programa:

Za ovaj dio programa želimo da se neprestano ponavlja. Kad slijed blokova naredbi završi, program se automatski ponavlja, za to će se pobrinuti naredba 'forever' iz kategorije 'Basic'. U prijašnjem dijelu programa smo postavili brojčanu vrijednost boje 255. Kako bismo bili u mogućnosti smanjiti jačinu boje, potrebno je tu vrijednost povećati ili smanjiti na nulu. Mi smo se odlučili za smanjivanje jačine svijetlosti zbog kontrasta svijetlo – tamno, prema tome ćemo postaviti našu vrijednost varijable na minus 1, kako bi se brojčana vrijednost 255, smanjivala za minus 1 dokle god ne dođe do nule. Na taj način, jačina svijetlosti će se ugasiti u potpunosti. Sljedeća naredbu koju pozivamo odnosi se na povezivanje naredbe 'set brightness item' iz kategorije naredbi LED.

Prilikom izvedbe ovog programa, dobit ćemo izgled mijenjanja kontrasta, a simbol koji smo odabrali će polako izblijediti na LED matrici.



Slika 61. Blok naredbi programa 'Svijetlo-tamno'

Mjerenje jačine magnetskog polja

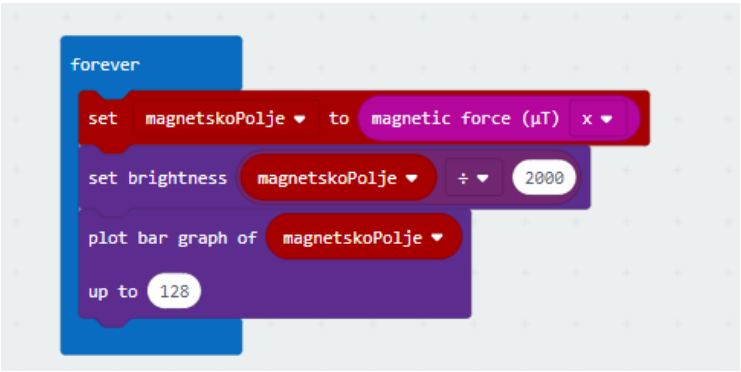
Cilj ovog zadatka je mjerenje jačine magnetskog polja. Zadatak započinjemo naredbom 'forever' što znači beskonačan broj ponavljanja zadatka. Iz kategorije Variables odabiremo naredbu 'set item to', no prije toga stvaramo novi 'item' pod nazivom 'magnetsko polje'. Iz kategorije Input odabiremo naredbu 'magnetic force' koja mjeri jačinu magnetskog polja. Nju postavljamo na vrijednost varijable 'to...'.

Jačinu polja ćemo mjeriti pojačavanjem razine svjetlosti, odnosno naredbom 'set brightness' iz kategorije LED. U taj blok naredbi prvo dodajemo naredbu iz kategorije Math koja označava računsku operaciju dijeljenja (Slika 62).

Slika 62. Postavljanje naredbe računske operacije dijeljenja

Nakon toga u prostor za upisivanje prvog broja, dodajemo varijablu 'magnetsko polje' metodom 'drag and drop', a vrijednost drugog broja postavljamo na 2000. Vrijednost 2000 određuje maksimalnu jačinu magnetskog polja.

Sljedeću naredbu koju ćemo dodati određuje minimalnu i maksimalnu vrijednost magnetskog polja. Naredba 'plot bar graph of' se sastoji od dvije brojčane vrijednosti. U prvu ubacujemo varijablu 'magnetsko polje', dok u drugu određujemo minimalni brojčani prikaz jačine od 128 (Slika 63). Prebacujemo program na micro:bit pločicu i možemo mjeriti jačinu magnetskog polja.



Slika 63. Program mjerenja jačine magnetskog polja

10. Zaključak

Kao što je na početku rada navedeno, pojam STEM povezuje znanost, tehnologiju i matematiku kao komponente pomoću kojih je korisniku omogućeno stvaranje programa kroz programiranje u micro:bit uređivaču. Način na koji se izražava ova vrsta tehnologije programiranja i njezina primjena u obrazovanju je izrazito kompleksno i neistraženo područje. Potrebno je osvrnuti se na eduakciju učitelja koji će ovo provoditi u školi. Primjenom programiranja u razrednoj nastavi, učenici će razvijati svoje logičko razmišljanje, a samim time i njihov opći uspjeh će rasti na svim područjima.

U Hrvatskoj je micro:bit novi pojam pomoću kojega će djeca naučiti razmišljati i povezivati na potpuno drugačiji način. Spoj kreativnost i programskog jezika je srž primjene micro:bita u razrednoj nastavi. U ovom radu sam povezala gradivo iz Nastavnog plana i programa sa slobodnim zadacima za uzrast djece razredne nastave pa i viših razreda. Primjena tih zadataka je povezana sa svim odgojnim i obrazovnim predmetima te informatike.

U današnje vrijeme informatička pismenost je dio opće kulture te na ovaj način možemo pridonijeti njezinom razvijanju od prvog razreda osnovne škole.

11. Literatura

1. https://www.srednja.hr/zbornica/nastava/evo-sto-znaci-i-koja-podrucja-obuhvaca-stem/ , pristupljeno 1.8.2018.
2. http://croatianmakers.hr/hr/stem-revolucija/ , pristupljeno 1.8.2018.
3. http://microbit.org/hr/about/, pristupljeno 1.8.2018.
4. http://microbit.org/hr/teach/, pristupljeno 1.8.2018.
5. https://e-radionica.com/hr/micro-bit-plocica.html, pristupljeno 3.8.2018.
6. Pandurić, T. (2017.) Uvod u micro:bit, OS Franje Krežme. Osijek
7. https://www.touchdevelop.com/microbit , pristupljeno 14.8.2018.
8. https://microbit.org/hr/guide/quick/ , pristupljeno 4.8.2018.
9. https://microbit.org/hr/2017-03-07-javascript-block-resources/ , pristupljeno 5.8.2018.
10. https://raf.edu.rs/citaliste/programiranje/4475-xa-sta-je-to-python-sve-sto-treba-znati-xa , pristupljeno 5.8.2018.
11. https://microbit-micropython.readthedocs.io/en/latest/tutorials/introduction.html , pristupljeno 6.8.2018.
12. https://microbit-micropython.readthedocs.io/en/latest/tutorials/hello.html , pristupljeno 6.8.2018.
13. http://izradi.croatianmakers.hr/project/mjerac-svjetlosti/ , pristupljeno 6.8.2018.
14. https://www.youtube.com/watch?v=fSlEmCTAxhE , pristupljeno 6.8.2018.
15. https://makecode.microbit.org/reference/basic , pristupljeno 10.8.2018.
16. https://makecode.microbit.org/reference/input , pristupljeno 10.8.2018.
17. https://makecode.microbit.org/reference/music , pristupljeno 11.8.2018.
18. https://makecode.microbit.org/reference/led , pristupljeno 11.8.2018.
19. https://makecode.microbit.org/reference/radio , pristupljeno 11.8.2018.