

Upotreba micro:bit uređaja u prikupljanju fizikalnih podataka

Bakula, Josip

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:357307>

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

Download date / Datum preuzimanja: **2024-10-04**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Josip Bakula

UPOTREBA MICRO:BIT UREĐAJA U
PRIKUPLJANJU FIZIKALNIH PODATAKA

Diplomski rad

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA I INFORMATIKA; SMJER: NASTAVNIČKI

Josip Bakula

Diplomski rad

**Upotreba micro:bit uređaja u
prikupljanju fizikalnih podataka**

Voditelj diplomskog rada: doc. dr. sc., Dalibor Paar

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2020.

Sažetak

U suvremenom obrazovanju, nastava fizike kao jedno od temeljnih STEM područja ima sve veći značaj. Ključna pitanja danas su kako učenike aktivno uključiti u nastavni proces, zainteresirati ih za fiziku i potaknuti dalje u smjeru STEM područja koja su okosnica zanimanja 21. stoljeća. O upotrebi mikroračunala u nastavi se već dugo razmišlja, no tek je prije 5 godina, konstrukcijom posebno dizajniranog mikroračunala BBC micro:bit, za potrebe obrazovanja u Europskoj uniji, došlo do masovne primjene u školama. Paralelno tome razvijaju se ideje i metode kako ih uključiti u nastavu pojedinih predmeta. Nastava fizike, koja u središtu ima fizikalni pokus, logično je mjesto implementacije mikroračunala. Micro:bit otvara mogućnost razvoja brojnih istraživačkih ideja u nastavi fizike uz relativno male troškove. Upotrebom micro:bit uređaja kao mjernog instrumenta u fizikalnom pokusu, uz mogućnost prikupljanja i bežičnog slanja podataka, učenike uvodimo u istraživački proces. Premda je preciznost mjerenja manja nego s profesionalnom opremom, ona je dovoljna za detekciju i analizu fizikalnih pojava. Micro:bitom izmjereni podaci prenose se na računalo, te se ovisno o razini obrazovanja na kojoj se tema obrađuje, mogu izvoditi jednostavne i složene analize i fizikalna modeliranja. Moglo bi se reći da je micro:bit i prvi korak u smjeru razvoja i dizajna novih uređaja ili visokotehnoloških proizvoda. U obrazovanju 21. stoljeća važno je da se učenici praktično upoznaju s tim procesom.

Ključne riječi: BBC micro:bit, nastava fizike, fizikalna mjerenja

The use of microbit devices in collecting physical data

Abstract

In modern education, the teaching of physics as one of the fundamental STEM areas is of increasing importance. The key questions today are how to actively involve students in the teaching process, increase their interest in physics and encourage them further in the direction of STEM areas that are the backbone of 21st century occupations. The use of microcomputers in teaching has been considered for a long time, but only 5 years ago the construction of a specially designed microcomputer BBC Micro:bit for the needs of education in the European Union led to mass application in schools. In parallel, ideas and methods are being developed on how to include them in the teaching of individual subjects. Teaching physics, which has a physical experiment at its center, is the logical place to implement microcomputers. Micro:bit opens the possibility of developing numerous research ideas in physics teaching at relatively low cost. By using micro:bit as a measuring instrument in a physical experiment with the possibility of collecting and sending data wirelessly, we introduce students to the research process. Although the measurement accuracy is lower than with professional equipment, it is sufficient for the detection and analysis of physical phenomena. Micro:bit-measured data are transferred to a computer and, depending on the level of education at which the topic is covered, simple and complex analyzes and physical modeling can be performed. It could be said that micro:bit is also the first step in the direction of development and design of new devices or high-tech products. In 21st century education, it is important that students become practically familiar with this process.

Keywords: BBC micro: bit, physics teaching, physical measurements

Sadržaj

1	Uvod.....	6
1.1	Uloga micro:bit uređaja u suvremenom obrazovanju.....	7
2	Micro:bit.....	9
2.1	Značajke micro:bit uređaja	9
2.2	Programiranje micro:bit uređaja	13
2.3	Povezivanje micro:bit uređaja radio vezom.....	15
2.3.1	Osnovni principi bežične komunikacije	15
2.3.2	Broadcast komunikacija micro:bit uređajem	18
2.3.3	Multicast komunikacija micro:bit uređajem.....	20
2.3.4	Unicast komunikacija micro:bit uređajem.....	21
2.3.5	Dvosmjerni unicast.....	22
3	Primjena u nastavi fizike	25
3.1	Prikupljanje podataka upotrebom micro:bit uređaja.....	25
3.2	Mjerenje temperature	28
3.3	Mjerenje ubrzanja i slanje podataka putem radio veze.....	31
3.3.1	Senzor zvučnih valova, detektor potresa	35
3.3.2	Matematičko njihalo	36
3.3.3	Micro:bit alarmni sustav	42
4	Zaključak.....	51
5	Literatura	52
6	Životopis.....	53

1 Uvod

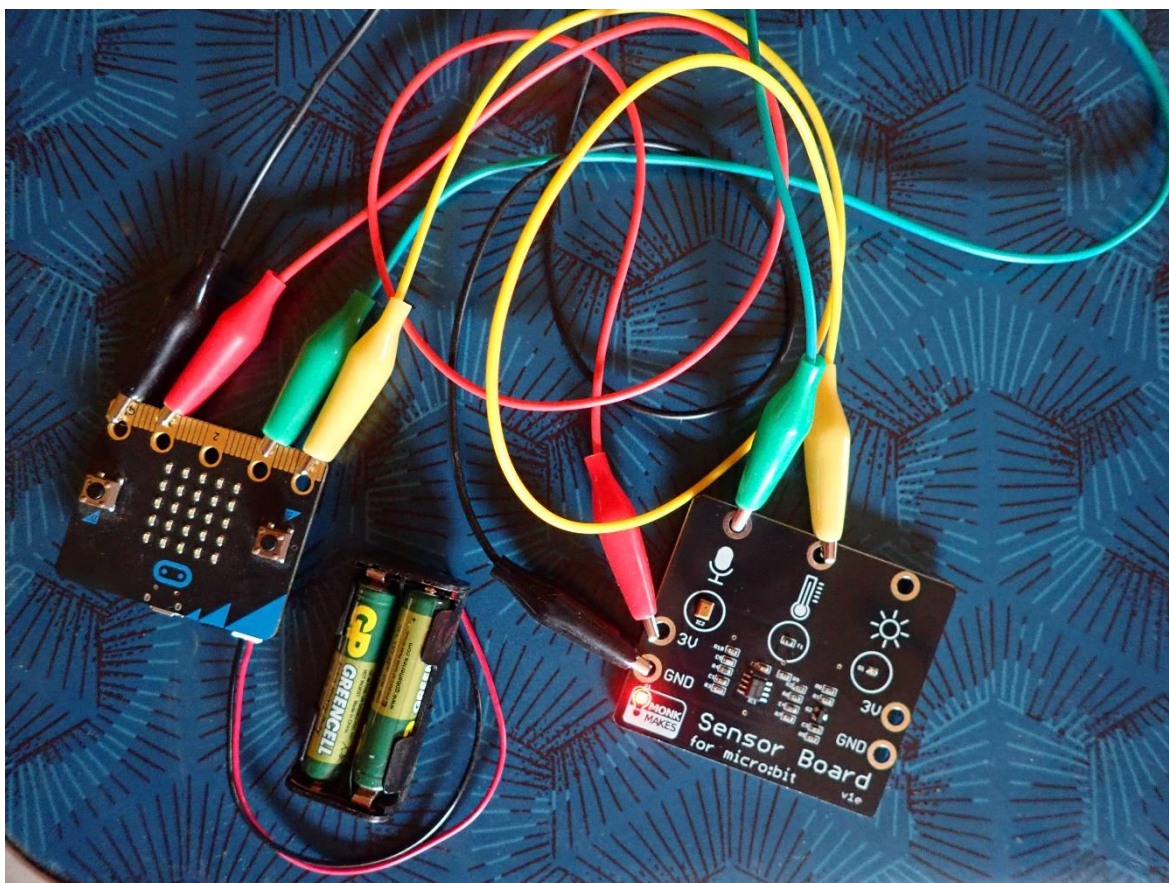
U izvedbi nastave fizike javljaju se mnoge poteškoće zbog kojih veliki broj nastavnika ne realizira temeljni dio sata fizike, pokus. Fizika je eksperimentalna znanost i sve zakonitosti moraju višestruko biti potvrđene pokusima. S druge strane, pokus predstavlja izvrstan alat za motivaciju učenika i aktivno uključivanje u nastavni proces, kroz istraživanja. Nastavnici ističu probleme kao što je nedostatak laboratorijske opreme, odnosno financijskih sredstava za opremanje fizikalnog kabineta. Zbog navedenog mnogi nastavnici preskaču konkretna mjerenja, na osnovu kojih učenici mogu praktično potvrditi zakone fizike, te se fokusiraju u većini na rješavanje zadataka, što bitno smanjuje motivaciju učenika i povezanost sa realnim svijetom.

Tehnološki laboratorij u nastavi fizike je realnost koju je moguće postići s malim financijskim ulaganjima, te izrazito dobro osmišljenim aktivnostima, kako bi se učenike maksimalno motiviralo da samostalno slažu mjerne postavbe i istražuju fizikalne zakonitosti. Danas, u doba izrazito povoljne potrošačke elektronike, možemo iskoristiti uređaje iz okoline kako bi analizirali i pojasnili fizičke pojave (Multiwingspan, n.d.) U sklopu rada, nastavnici imaju priliku upoznati se s mikroračunalom kao pomagalom u nastavi fizike, testirati senzore, obraditi dobivene podatke te samostalno osmisliti pokus kojim bi na jednostavan način mjerili fizičke veličine u svojoj okolini (intenzitet svjetlosti, intenzitet zvuka, temperaturu, vlažnost zraka, gibanje, magnetsko polje te mnoge druge).

Ideja ovog rada je da predstavimo neke od primjera primjene mikroračunala u nastavi fizike. Posebno ćemo naglasiti činjenicu da se koristeći mikroračunala mogu dizajnirati brojni istraživački postavi, kojima se u svrhu istraživanja pojedinih fizikalnih pojava izvode kvantitativna mjerenja. Izmjereni podaci se prenose na računalo, te se ovisno o razini obrazovanja na kojoj se tema obrađuje, mogu izvoditi jednostavne i složene analize, pa i fizikalna modeliranja. Pored micro:bita, koji je dizajniran za rano učenje STEM vještina u osnovnoj školi, u srednjoj školi i na fakultetima mogu se koristiti i kompleksnije arhitekture, poput Arduina i Raspberry pi, koje otvaraju mogućnost i profesionalnih primjena (mjerni instrumenti i senzori bazirani na tim arhitekturama).

1.1 Uloga micro:bit uređaja u suvremenom obrazovanju

Micro:bit je mikroročunalo posebno razvijeno za potrebe obrazovanja. Na prvi pogled ne izgleda spektakularno, no ima značajnu razliku u odnosu na sve vrste računala kojima smo okruženi. Njegova arhitektura vidljiva je i dostupna učenicima, već uz kratke upute na njega se mogu spajati vanjski senzori ili motori, i za razliku od mobitela i klasičnih računala, ono omogućava učenicima da sami razvijaju interakciju između računala i svijeta koji ih okružuje. Mogućnost intuitivnog i jednostavno učenja kako programirati micro:bit i spajati vanjske senzore (Slika 1) i uređaje čini ovo mikroročunalo idealnim alatom za početne korake u STEM obrazovanju u osnovnoj školi, uz brojne moguće interdisciplinarne primjene, posebice u nastavi fizike. Unatoč skromnih hardverskim mogućnostima, micro:bit može biti moćan alat u obrazovnom procesu, posebice u svim područjima prirodoslovlja, gdje se može koristiti kao instrument za prikupljanje kvantitativnih podataka u različitim prirodnim procesima i pokusima u razredu.



Slika 1 Primjer micro:bit uređaja povezanog vanjskim senzorima

Micro:bit danas predstavlja primjer korištenja suvremene tehnologije u nastavnom procesu (u obrazovanje u Europskoj uniji se počeo uvoditi 2015. godine). Prednosti korištenja ove tehnologije su višestruke, uređaji omogućuju brzo učenje osnovnih funkcija,

što je bitno za motivaciju, jednostavno se nadograđuju vanjskim sensorima i upravljačkim komponentama i time lako integriraju u različite projekte u nastavi, učenike uvodimo u stvarno mjerenje u vrlo ranoj dobi, analiziramo mjerene veličine, razvijamo analitičko mišljenje, procjenjujemo rezultat, interaktivnost nastavnog procesa dovodimo do krajnjih granica, motiviramo učenike, te im pružamo uvid u uređaj koji će moći samostalno koristiti kod kuće u svakodnevnom životu i budućnosti.

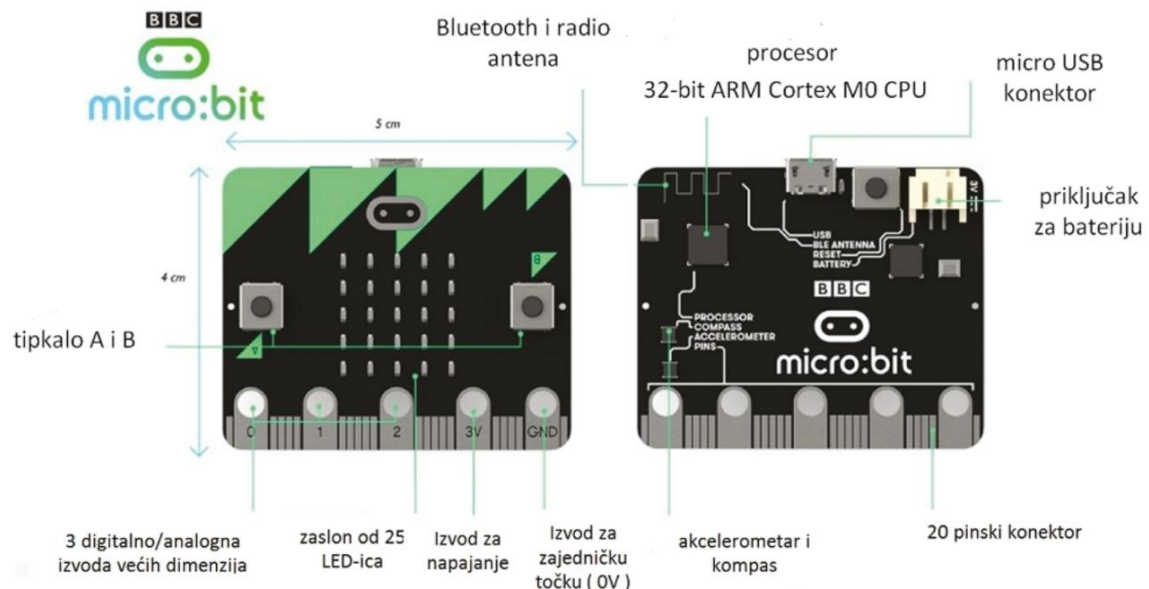
Mlade osobe sa stečenim sveobuhvatnim poimanjem svijeta, koji poznaju statističku obradu, barataju programskim jezicima, te mogu svoja znanja primijeniti u novim okolnostima, biti će stup nositelj budućeg društva. Takvi mladi ljudi imati će priliku birati pravac napredovanja u profesionalnom smislu te upravljati najbitnijim odlukama u društvu. Zbog toga je važnost integracije programiranja te mikroračunala ključna u razvoju buduće edukacije svih učenika RH (Pintarić, 2017).

Istaknimo i da nakon 5 godina primjene u obrazovanju, u listopadu 2020. micro:bit doživljava prvu veliku hardversku nadogradnju na *64MHz ARM procesor* (prethodno je bio 16MHz), 8 puta veći RAM (*120kB*) i dvostruki iznos fleš memorije (*512kB*). Novi model ima ugrađen mikrofon i zvučnik, kao i podršku za aplikacije koje su bazirane na umjetnoj inteligenciji, što otvara mogućnosti za brojne nove potencijalne primjene u obrazovanju.

2 Micro:bit

2.1 Značajke micro:bit uređaja

U ovom poglavlju dati ćemo pregled nekih od osnovnih značajki micro:bita, s naglaskom na one koje su posebno korisne za upotrebu u nastavi fizike. Micro:bit je mikroracunalo koje možemo programirati, upravljati njegovim sučeljem i sensorima. Jedna od izuzetnih mogućnosti micro:bita je relativno lako spajanje vanjskih senzora i kontrolera (npr. za upravljanje elektromotorima i prekidačima, što otvara široke primjene). Zbog toga ga se ponekad naziva *mikrokontroler*. Osnovni dijelovi (Slika 2) su mu procesor, zaslon s diodama, ulazni i izlazni izvodi, interni senzori, bežična veza i napajanje (baterije ili USB kabel, kojim se napaja iz računala ili vanjskog napajanja).



Slika 2 Prikaz micro:bit mikroracunala

Procesor

Procesor upravlja mikroracunalom izvršavajući naredbe napisane u odgovarajućem programskom kodu (MakeCode, Python, JavaScript). Svojstva novog procesora, koji kreće u upotrebu krajem 2020. godine, dana su u tablici 1.

<i>Model</i>	<i>Nordic nRF52833</i>
<i>Core variant</i>	<i>Arm Cortex-M4 32 bit processor with FPU</i>
<i>Flash ROM</i>	<i>512KB</i>
<i>RAM</i>	<i>128KB</i>
<i>Speed</i>	<i>64MHz</i>

Tablica 1

Zaslon sa svijetlećim diodama

Micro:bit s prednje strane ima zaslon koji sadrži 25 svijetlećih dioda, koje se mogu pojedinačno programirati, što predstavlja izvrstan alat za učenje principa funkcioniranja sučelja i računalne grafike. Svaka dioda ima svoju poziciju određenu koordinatama na x i y osi. Programiranjem svijetlećih dioda na zaslonu se mogu prikazati brojke, tekst, sličice i animacije. Svijetleće diode služe i kao senzor za svjetlo, što možemo upotrebiti kao svjetlosni prekidač u fizikalnim pokusima.

Na tom relativno jednostavnom ekranu mogu se prikazivati podaci i time izvoditi osnovna komunikacija s korisnikom. Ako se radi o većem broju podataka princip rada je da se oni prebace (bežičnom vezom) na računalo ili mobitel i tamo dalje obrađuju.

Tipkala A i B

Micro:bit uređaj s prednje strane sadrži dva tipkala, A i B. Tipkala svako zasebno, i u kombinaciji, omogućuju upravljanje informacijama koje su pohranjene na micro:bit uređaju, kao i pokretanje ranih dijelova programskog koda. To nije jedini način interakcije s korisnikom. Jedan od njih je promjena orijentacije uređaja, uz pomoć akcelerometra, čime se detektiraju različiti položaji ili načini kretanja.

Tipkalo za ponovno pokretanje

Sa stražnje strane micro:bit-a nalazi se tipkalo za ponovno pokretanje. Pritiskom na tipkalo, program koji se nalazi na micro:bit uređaju pokreće se ispočetka.

Micro USB konektor

Kako bi izrađeni program mogli prenijeti na micro:bit uređaj, potrebno je povezati micro:bit s računalom pomoću mikro USB kabela. Povezivanjem s računalom micro:bit se napaja električnom energijom, zbog čega je moguć rad bez baterije.

Kompas

Kompas na micro:bit uređaju omogućuje detektiranje magnetskog polja Zemlje, te smjera prema kojem je micro:bit uređaj okrenut (azimuta). Azimut se mjeri u stupnjevima. Prije uporabe potrebno je kalibrirati kompas.

Akcelerometar

Mjerač ubrzanja (akcelerometar) je dio micro:bit-a koji možete koristiti kao senzor pokreta. Prepoznaje kada se micro:bit kreće (gibanje, drhtanje, nagib i slobodni pad)

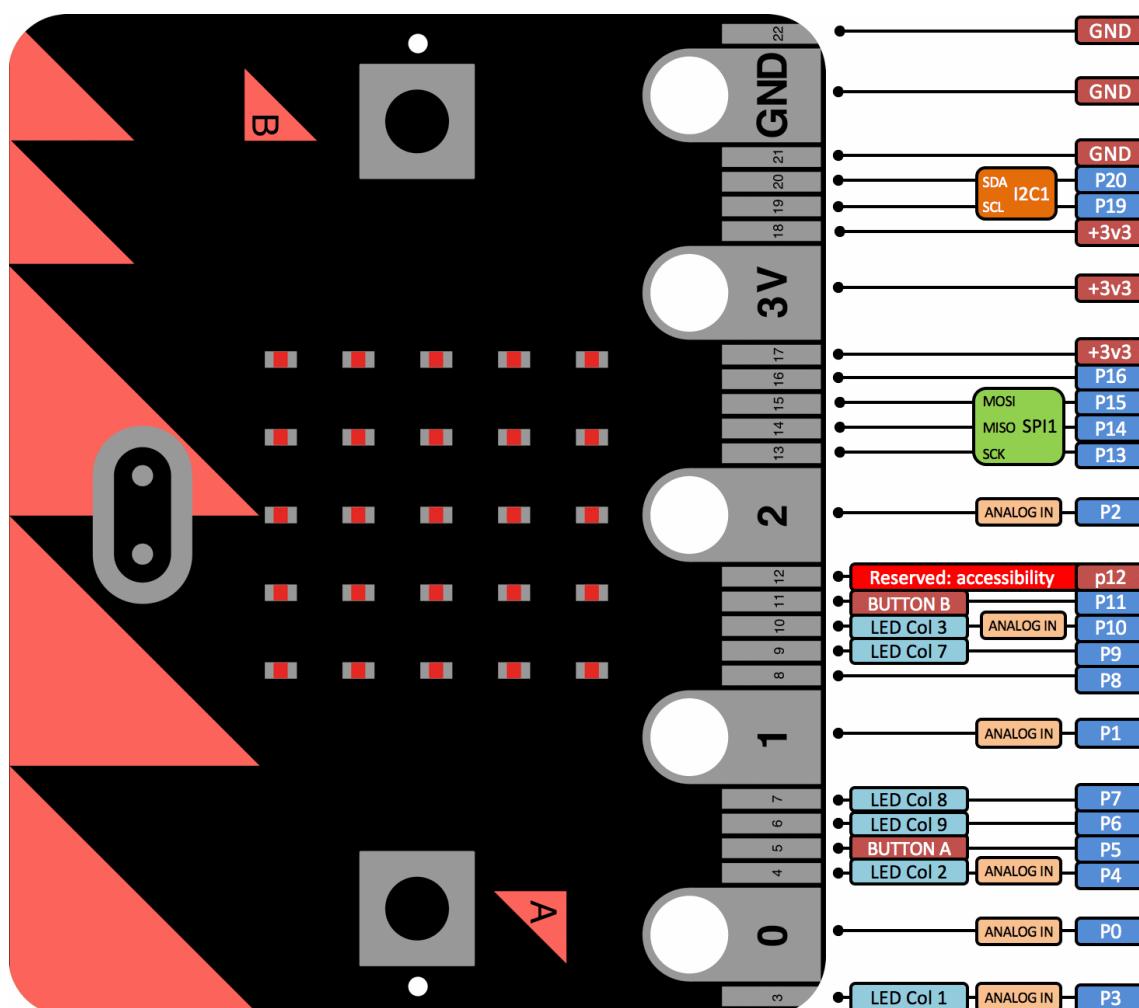
Temperaturni senzor

Najveća temperatura koju može izmjeriti i prikazati iznosi 50°C. Senzor mjeri temperaturu micro:bita, na koju utječe i sam uređaj, tako da je za praćenje temperature u pokusima bolje koristiti vanjske senzore. Interni temperaturni senzori u računalima najčešće

imaju ulogu kontrolne provjere radne temperature, kako bi se izbjeglo moguće pregrijavanje i kvar elektronike, što se može dogoditi ako uređaj tijekom ljeta direktno izložimo sunčevom zračenju.

Izvodi

Micro:bit na rubnom dijelu ima ukupno 25 vanjskih izvoda (Slika 3). Kontrola tih izvoda predstavlja jedan od najmoćnijih dijelova ovog mikroračunala, jer omogućava dvosmjernu (i relativno jednostavnu) komunikaciju i interakciju s okolinom. Prva tri veća izvoda (0, 1 i 2) mogu imati ulogu ulaznih ili izlaznih jedinica, te na njih, zbog svoje veličine, možemo lako spojiti dodatne komponente. Izvod 3V je napajanje, a GND je izvod za uzemljenje. Ostalih 20 malih izvoda možemo koristiti za komunikaciju i upravljanje s dodatnim komponentama.



Slika 3 Vanjski izvodi micro:bit uređaja

Komunikacija radio vezom

Micro:bit sadrži radio komponentu za bežičnu komunikaciju s drugim micro:bit uređajima. Povezivanjem više micro:bit uređaja radio vezom moguće je izraditi kompleksnije projekte (primjerice daljinski upravljači, uređaji za sudjelovanje u kvizu, igraće konzole i sl.). Jedna od najzanimljivijih primjena za obrazovanje je prijenos podataka, gdje micro:bit, koji prikuplja podatke, šalje prikupljene podatke drugom micro:bitu koji je spojen na računalo. Druga opcija je direktna komunikacija između micro:bita i prijenosnih računala putem bluetootha.

<i>Stack</i>	<i>Bluetooth 5.1 with Bluetooth Low Energy(BLE)</i>
<i>Band</i>	<i>2.4GHz ISM (Industrial, Scientific and Medical) 2.4GHz..2.41GHz</i>
<i>Channels</i>	<i>50 2MHz channels, only 40 used (0 to 39), 3 advertising channels (37,38,39)</i>
<i>Sensitivity</i>	<i>-93dBm in Bluetooth low energy mode</i>
<i>Tx Power</i>	<i>-40dBm to 4dBm</i>

Tablica 2

<i>Protocol</i>	<i>Micro:bit Radio</i>
<i>Freq band</i>	<i>2.4GHz</i>
<i>Channel rate</i>	<i>1Mbps or 2Mbps</i>
<i>Encryption</i>	<i>None</i>
<i>Channels</i>	<i>80 (0..80)</i>
<i>Group codes</i>	<i>255</i>
<i>Tx power</i>	<i>Eight user configurable settings from 0(-30dbm) to 7 (+4dbm)</i>
<i>Payload size</i>	<i>32 (standard) 255 (if reconfigured)</i>

Tablica 3

2.2 Programiranje *micro:bit* uređaja

Micro:bit se može programirati u više besplatno dostupnih programskih okruženja, poput MakeCodea, Pythona i JavaScripta. U ovom radu prikazani su primjeri programiranja pomoću alata *Microsoft MakeCode*, prikladnog za početno učenje. Microsoft MakeCode je grafičko sučelje unutar kojeg se programski kôd izrađuje povlačenjem i ispuštanjem blokova kôda, koji se međusobno spajaju. Također je moguće umjesto blokovskih naredbi koristiti Python ili JavaScript programski jezik. U nekim primjerima potrebno je umjesto online alata MakeCode koristiti aplikaciju *MakeCode for micro:bit* za Windows 10. Kod napravljen u MakeCode alatu moguće je konvertirati u Python ili JavaScript.

Za početak rada potrebno je imati:

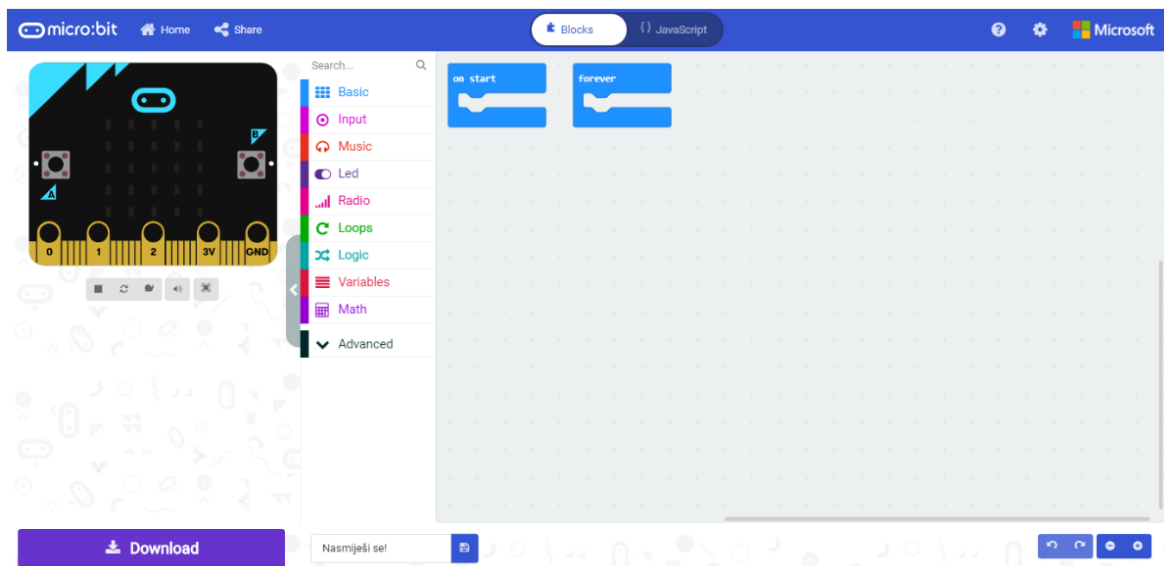
- *micro:bit* uređaj, s time da rezultate programiranja možete provjeriti na simulatoru u MakeCode sučelju, dostupnom na makecode.microbit.org
- mikro USB kabel kojim se *micro:bit* fizički povezuje s osobnim računalom na kojem programirate,
- računalo,
- pristup internetu.

Izrada programa u MakeCode programskom okruženju

Kako biste mogli započeti sa izradom programa, otvorite stranicu makecode.microbit.org, te otvorite *novi projekt* unutar Microsoft MakeCode programskog okruženja, klikom na stavku *New Project*. Na ovoj stranici možete pristupiti svim svojim projektima vrlo jednostavno i brzo bez dodatnog pohranjivanja programa na vlastito računalo. Također, možete učitati već izrađene programe klikom na *Import*.

Prilikom otvaranja novog projekta, blokovi *on start* i *forever* već će biti u prostoru za programiranje (Slika 4). S lijeve strane nalazi se simulator *micro:bita*. Na simulatoru se prikazuje trenutni program kojeg izrađujete, te možete odmah uočiti promjene prilikom izrade programa. Osim pokretanja i zaustavljanja programa na simulatoru, pritiskom na ikonu puža ispod njega, program se sporije izvršava te se pritom istovremeno u sivom prostoru ističu naredbe koje čine taj program. Time se dobiva uvid na koji način pojedina naredba programa radi, kao i bolje razumijevanje slijednog izvršavanja naredbi prilikom programiranja.

U sredini je izbornik s glavnim kategorijama naredbi koje se koriste u programiranju. Klikom na svaku od ovih kategorija otvara se popis naredbi u obliku blokova, te se one povlačenjem mišem smještaju u prostor za slaganje naredbi. S desne strane nalazi se sivi prostor u kojem stvaramo program povlačenjem i ispuštanjem blokova kôda koji se međusobno spajaju. Kada je program završen, preuzme se na računalo i USB kablom prenese na micro:bit.



Slika 4 Micro:bit MakeCode radni prostor (makecode.microbit.org)

2.3 Povezivanje micro:bit uređaja radio vezom

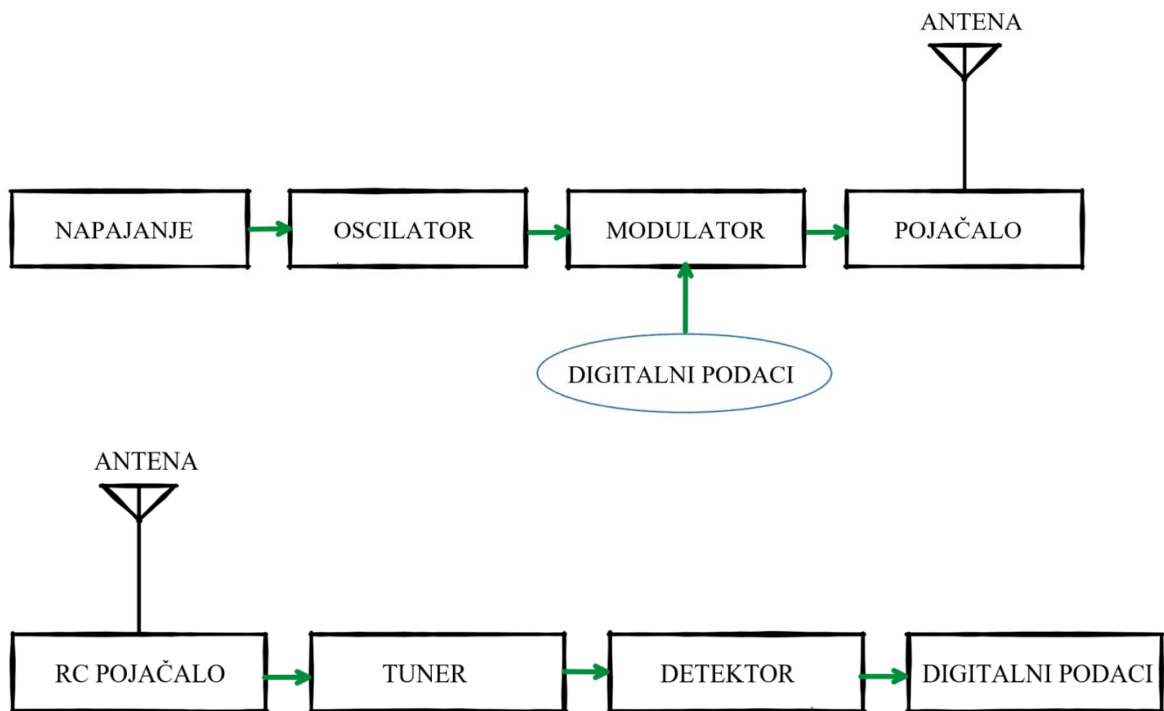
2.3.1 Osnovni principi bežične komunikacije

Bežična komunikacija danas postaje standard u svakodnevnoj uporabi, od mobitela, računala koja se bežično spajaju na internet do *IoT* (*Internet of things* – brojnih drugih uređaja, vozila, industrijskih komponenti, senzora koji se spajaju na internet i razmjenjuju podatke).

Bežična komunikacija fizikalno funkcionira pomoću elektromagnetskih valova, frekvencija koje uključuju područja radiovalova i mikrovalova (3 kHz – 300 GHz).

U osnovama, svaki radio odašiljač ima napajanje, oscilator, modulator koji uključuje digitalni signal koji šaljemo, pojačalo i antenu, koji zajedno generiraju radio valove koji putuju do prijemnika (Slika 5).

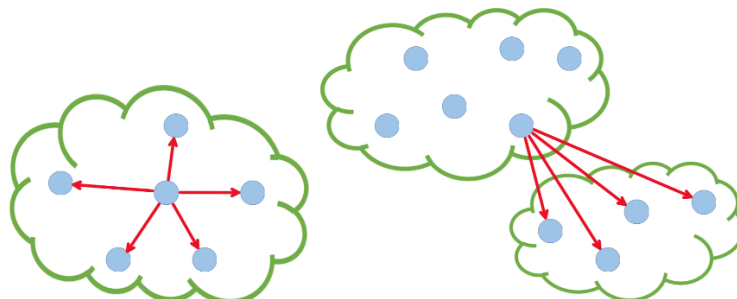
Antena odašiljača omogućava da se energija (u obliku elektromagnetskog vala) šalje u prostor, a antena prijemnika prima tu energiju. Prijemnik se sastoji od antene, RF pojačala, tunera, i detektora iz kojeg izlazi primljeni digitalni signal.



Slika 5 Shematski prikaz odašiljača i prijemnika

Ovisno o zahtjevima, potrebama, ograničenjima mreže i upotrebljenim protokolima, razlikujemo više vrsta bežične komunikacije putem radiovalova, kao što su:

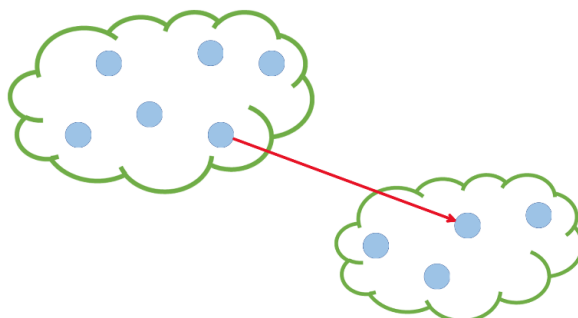
Broadcast komunikacija – podrazumijeva da se poruka jednog pošiljatelja prenosi svim primateljima unutar mreže (Slika 6).



Slika 6 Ograničeni broadcasting i direktni broadcasting

Multicast komunikacija – poruka jednog ili više pošiljatelja prenosi se jednom ili više primatelja unutar mreže.

Unicast komunikacija – poruka jednog pošiljatelja prenosi se jednom primatelju unutar mreže (Slika 7).



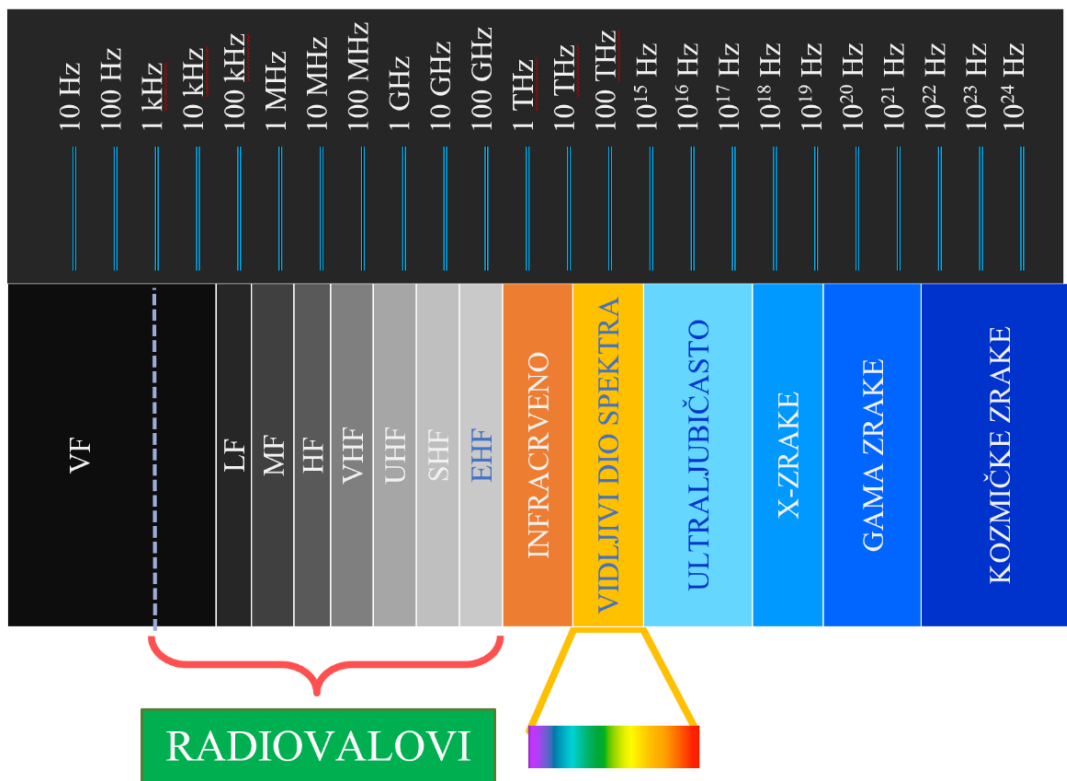
Slika 7 Unicast komunikacija

Prije no zakoračimo u područje informacijsko komunikacijskih tehnologija, poželjno je produbiti razumijevanje fundamentalnih zakona prirode, zaslužnih za uspješan prijenos podataka bežičnim putem.

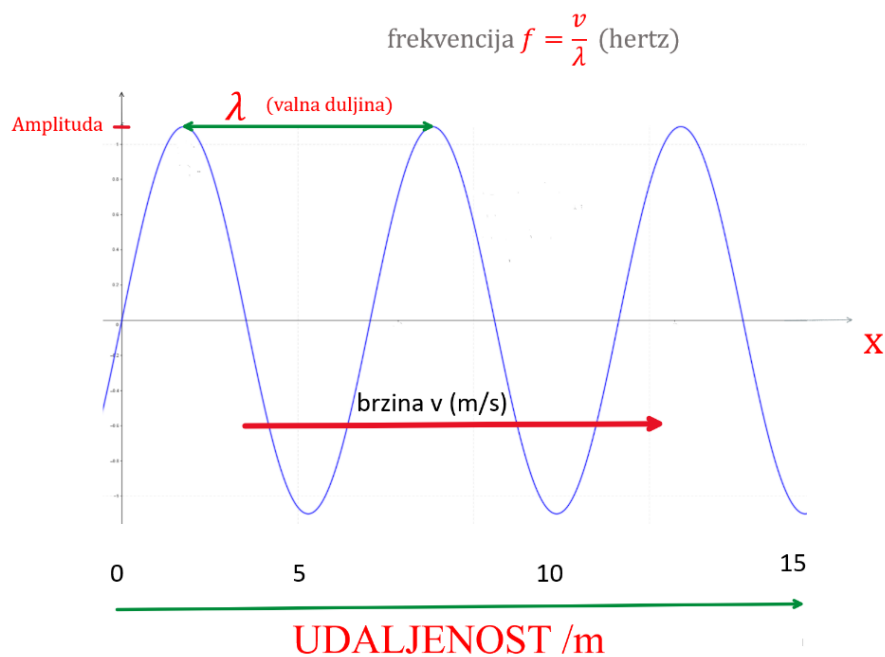
Pozadina radio komunikacije: EM valovi

Radiovalovi su elektromagnetski valovi i predstavljaju dio spektra elektromagnetskih valova određenog frekventnog područja, od 3 kHz – 300 GHz (Slika 8). Elektromagnetski val općenito definiran je s tri veličine, valnom duljinom, frekvencijom i brzinom (Slika 9). U vakuumu elektromagnetski valovi šire se brzinom svjetlosti $c = 299$

792 km/s (u zraku ta brzina je nešto manja ali i dalje vrlo blizu brzini svjetlosti). Micro:bit komunicira u frekventnom području 2402 MHz - 2480 MHz.



Slika 8 Spektar elektromagnetskog zračenja



Slika 9 Valna duljina i njen odnos sa frekvencijom i brzinom širenja vala

2.3.2 Broadcast komunikacija micro:bit uređajem

Broadcast komunikacija podrazumijeva da se poruka jednog pošiljatelja prenosi svim primateljima unutar mreže. Broadcast adresa je posebna adresa koja omogućuje zaprimanje poruka svim uređajima unutar mreže, odnosno govori da poruku trebaju vidjeti svi uređaji unutar iste mreže

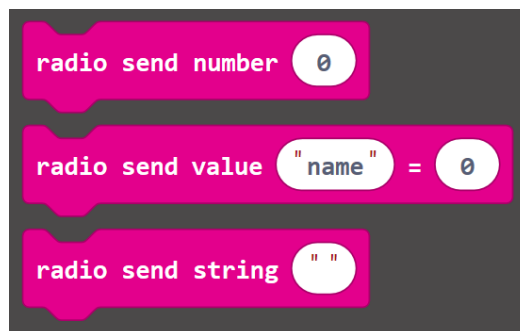
Postavljanje radija

Da bismo povezali dva (ili više) micro:bit uređaja putem radiovalova nužno je da svi uređaji koriste istu *radio grupu*, koja će se koristiti kao broadcast adresa (Runberg, n.d.). Za postavljanje radio grupe, odnosno broadcast adrese, potrebno je koristiti funkciju *radio.set_group()*, kojom se ID radio grupe može postaviti na vrijednost od 0 do 255.

Programski kod potreban za komunikaciju dva micro:bit uređaja razlikuje se ovisno koristi li se uređaj kao odašiljač podataka ili prijemnik. Micro:bit uređaj ima mogućnost pomoću radiovalova emitirati brojeve i znakovne nizove, koristeći odgovarajuće funkcije.

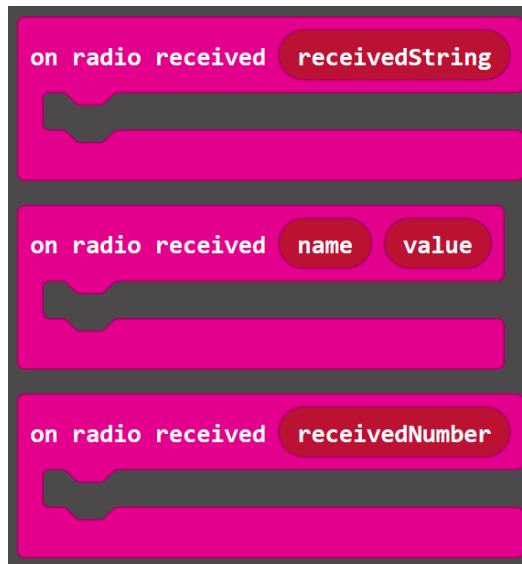
Komunikacija brojčanim vrijednostima

Za emitiranje brojčanih vrijednosti potrebno je koristiti funkciju *radio.send_number()*. Paketi podataka koji su znakovnog tipa emitiraju se funkcijom *radio.send_string(" ")*, a moguće je slati i pakete koji sadrže takozvane parove podataka funkcijom *radio.send_value(" ",)*, koja emitira po jednu brojčanu i znakovnu vrijednost (Slika 10). Maksimalna duljina 'stringa' je 12 znakova.



Slika 10 Primjeri slanja radio komunikacijom različitih tipova podataka (Microsoft MakeCode)

Micro:bit uređaj koji prima paket podataka koristi funkcije *on_recieved_number()*, *on_recieved_string()*, ili *on_recieved_value()*, ovisno o tipu podataka koji je emitiran (Slika 11).

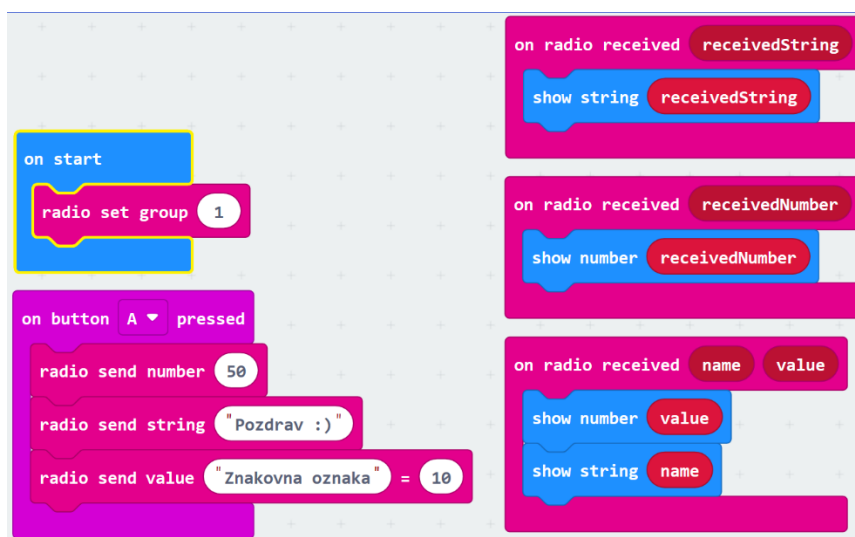


Slika 11 Primanje radio komunikacijom različitih tipova podataka

Primjer programa za slanje podataka putem radio komunikacije

Program za dvosmjernu radio komunikaciju (Slika 12) demonstrira vrste podataka koje je moguće slati putem radio veze, a programski kod u Pythonu s istom funkcionalnosti izgledao bi ovako:

```
def on_button_pressed_a():
    radio.send_number(50)
    radio.send_string("Pozdrav")
input.on_button_pressed(Button.A, on_button_pressed_a)
```



Slika 12 Program za dvosmjernu radio komunikaciju

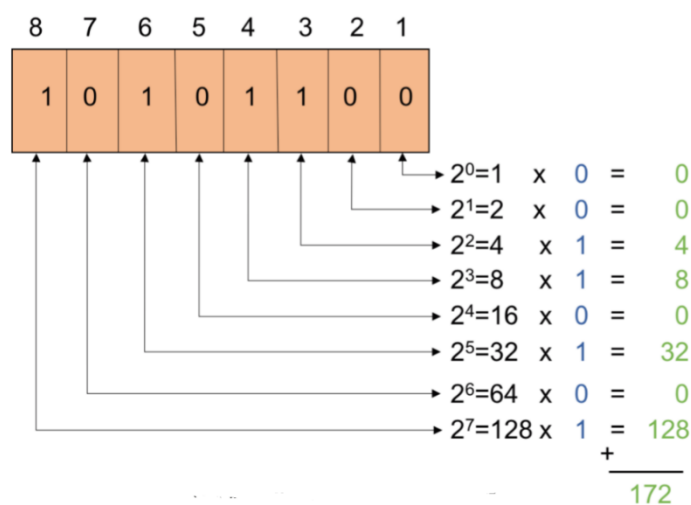
2.3.3 Multicast komunikacija micro:bit uređajem

Program sa Slika 12 primjer je broadcast komunikacije, što podrazumijeva da uređaj emitira poruke svim uređajima u blizini. Ako na svaki micro:bit uređaj prenesemo kod za odašiljanje i primanje podataka, te pokušamo testirati komunikaciju u učionici s velikim brojem micro:bit uređaja, susresti ćemo se s mnoštvom poslanih i zaprimljenih poruka koje mogu otežati komunikaciju.

U slučaju kada se koristi veliki broj micro:bit uređaja, često je prikladno slati podatke samo nekima od njih, odnosno učenike bi trebalo podijeliti na manje grupe, koje će upotrebom dva ili više uređaja međusobno komunicirati. Ograničavanjem uređaja na način da mogu slati poruke samo uređajima unutar svoje grupe, odnosno da mogu prihvatiti samo poruke poslano od uređaja iz iste grupe, dobijemo *multicast komunikaciju*, kakva se koristi u mrežnim tehnologijama, kada je putem interneta potrebno slati podatke prema odabranim korisnicima u isto vrijeme, primjerice za vrijeme videokonferencija, ili za potrebe internetske televizije.

Za grupnu komunikaciju micro:bit uređajima poslano poruke je potrebno označiti, odnosno pridružiti im *grupnu/multicast adresu*, ili *grupni ID*. Grupna adresa/ID omogućuje ostalim uređajima da prepoznaju kojoj grupi emitirana poruka pripada, i tako odluče trebaju li prihvatiti poruku.

Pri postavljanje grupne adrese i u ovom slučaju koristi se *radio.set_group()* funkcija, no ovaj put micro:bit uređaji iz različitih grupa ne smiju imati istu adresu. Vrijednosti koje možemo pridružiti grupnoj adresi u rasponu su od 0 do 255. Zapis vrijednosti grupne adrese u binarnom sustavu čini točno 8 bitova (Slika 13).



Slika 13 Primjer 8-bitne adrese.

2.3.4 Unicast komunikacija micro:bit uređajem

Unicast komunikacija („jedan-na-jedan“) podrazumijeva slanje podataka jednom primatelju. Tipičan je način kojim komuniciramo putem interneta, npr. da bismo otvorili i vidjeli mrežnu stranicu, potrebno je poslati zahtjev poslužitelju, koji nam šalje web stranicu za prikaz u našem mrežnom pregledniku.

Kada se prenose podaci samo prema jednom primatelju, uređaji koji izmjenjuju podatke koriste *protokole*. Pojednostavljeno, protokoli definiraju kako uređaji trebaju slati podatke i što trebaju učiniti sa zaprimljenim podacima. Promatrajući internet, svaki povezani uređaj prati internetski protokol (IP), prema kojem je svakom uređaju dodijeljena jedinstvena IP adresa.

IP adrese (korištene kao unicast adrese na internetu) čini znakovni niz, kojeg čine do 4 decimalna broja odvojena točkom, u rasponu vrijednosti od 0 do 255 (npr. 161.53.55.3). Micro:bit uređajima su, pri bežičnoj komunikaciji, također dodijeljene adrese, koje smo djelomično kontrolirali mijenjajući ID grupe.

Dva računala/uređaja komuniciraju putem *paketa podataka*, sastavljenih od sadržaja poslanog dijela poruke, i jednog ili više *zaglavlja*, koji sadrže korisne informacije za protokole, poput IP adresa primatelja i pošiljatelja.



Slika 14 Podatkovni paket

Slika 14 prikazuje strukturu paketa podataka. Osim adresa primatelja i pošiljatelja, zaglavlje sadrži informaciju o tipu podataka koji se šalje paketom, npr. tekst ili slika.

Upotreba različitih protokola može povlačiti dodavanje različitog zaglavlja sadržaju koji se šalje.

Za unicast komunikaciju među micro:bit uređajima, potrebno je prirediti paket dodavajući zaglavlje sa adresama pošiljalca i primatelja. Da bi unicast funkcionirao, radio micro:bit uređaja treba primiti sve poslone poruke, ali pročitati samo one koje su adresirane na njega, na sličan način na koji u jedno domaćinstvo može pristizati više pisama, od kojih bismo otvorili samo ona koja su upućena nama osobno.

Za pristizanje svih paketa, poslanih od bilo koga, potrebno je kao podlogu koristiti broadcast komunikaciju. Nakon postavljanja radio grupe, svaki micro:bit pošiljalac treba svakoj emitiranoj poruci dodati zaglavlje, koje sadrži adrese primatelja i pošiljalca, koristeći u obliku znakovnog niza.

Prva dva znaka definirat će adresu. Svaki niz dodjeljuje se samo jednom micro:bit uređaju unutar istog dosega. Dodijeljene znakovne nizove (npr. AB i BC) za pošiljalca i primatelja potrebno je povezati da bi se stvorilo zaglavlje. Povezanim adresama pošiljalca i primatelja potrebno je dodati sadržaj koji se šalje.

```
radio.send_string("AB" + "CD" + "Ugodan dan!")
```

Primatelju su nužne informacije iz zaglavlja kako bi mogao donijeti odluku o primitku poruke. Za provjeru nalazi li se u zaglavlju poruke adresa vlastitog uređaja, primatelj treba razdvojiti zaprimljeni paket u znakovne nizove s adresom pošiljalca, adresom primatelja i jedan sadržajem poruke naredbom, npr. *substring*, a zatim provjeriti podudara li se adresa primateljevog uređaja s adresom primatelja koja je navedena u zaglavlju, koristeći *compare()*.

```
posiljalac = podatkovni_paket.substr(0, 2)
primatelj = podatkovni_paket.substr(2, 2)
sadržaj = podatkovni_paket.substr(4, 8)
posiljalac.compare("AB")
primatelj.compare("CD")
```

U slučaju da su spomenute adrese identične, uređaj koji je zaprimio paket zaista je odgovarajući primatelj.

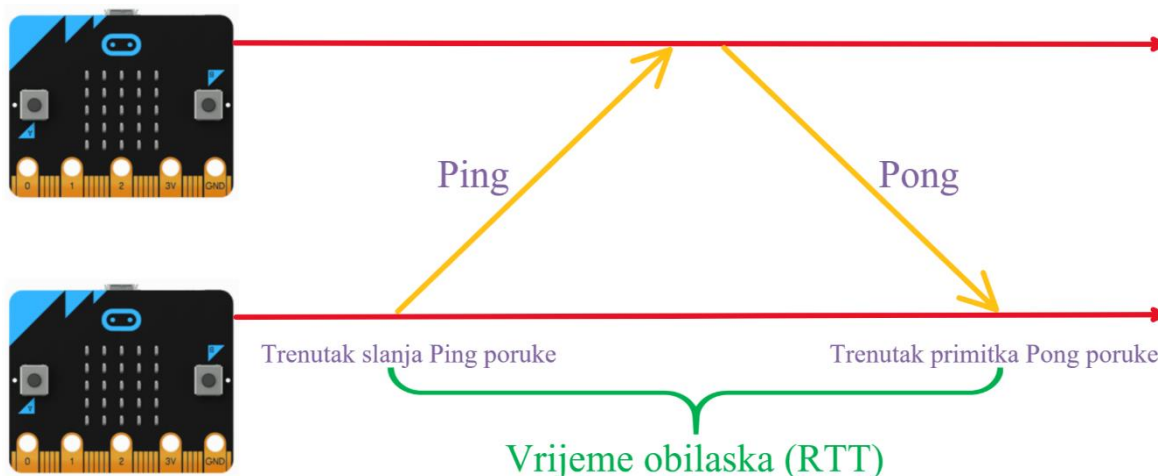
Postoje slučajevi kada ne želimo na uređaj primiti sve poruke poslone s bilo kojeg uređaja. Zadovoljiti takve uvjete možemo uspoređujući i adresu pošiljalca, iz zaglavlja poslanog paketa, s adresama primatelja kojima je dozvoljeno slanje poruka.

2.3.5 Dvosmjerni unicast

Slanje podataka na drugi micro:bit i dobivanje povratnog odgovora omogućeni su dvosmjernom komunikacijom, kojom se podaci prenose u oba smjera, ali ne nužno u isto vrijeme.

U većini slučajeva svaki micro:bit uređaj ima svoju ulogu ovisno koristi li se za emitiranje ili prijem signala. Dvosmjerna komunikacija svakom uređaju omogućuje slanje i primanje podataka, čime se otvara mogućnost kreiranja dvosmjernih protokola. U takvim se protokolima, nakon što računalo/uređaj pošalje paket podataka, čeka određen odgovor na poslan paket.

Često korišten primjer dvosmjernog protokola, za provjeru je li uređaj uključen i povezan mrežom, jeste ping program. Ping program šalje „Ping“ poruku, i očekuje da će poslana poruka odjeknuti nakon što stigne do primatelja, za primjer „Pong“ porukom. Slično kao neka vrsta ping-pong igre, koju igraju računala putem mreže. Ukoliko pošiljatelj ne primi povratnu poruku kao odgovor na poslani „Ping“, ili je potrebno dulje vrijeme čekati na povratni „Pong“, može se zaključiti da postoji problem s vezom. Ping program mjeri takozvano *vrijeme obilaska*, ili *RTT (Round trip time)*, koje je potrebno za oba prijenosa podataka. Drugim riječima, pošiljatelj mjeri razliku u vremenu između poslanog Ping-a i primljenog Pong-a (Slika 15).



Slika 15 Vrijeme obilaska (RTT)

Za mjerenje vremena obilaska micro:bit uređajem koristi se *running time* varijabla, koja bilježi vrijeme proteklo od posljednjeg uključivanja/ponovnog pokretanja micro:bita, u milisekundama. Potrebno je započeti s unicast komunikacijom, te odlučiti koji uređaj će slati Ping poruke, a koji će odgovarati Pong porukom, kako bi se moglo dizajnirati zaglavlje, Ping paket, i Pong paket.

Pošiljalac ping poruke bilježi trenutnu vrijednost *running time* prije slanja poruke, kako bi obilježio trenutak i započeo mjerenje vremena potrebnog za prijem povratne poruke. Primitkom ping poruke, drugi micro:bit odgovara porukom pong. Zaprimanjem Pong poruke, prvi micro:bit uređaj bilježi taj trenutak i može započeti računanje rtt vremena, koje je jednako razlikama zabilježenih vrijednosti varijable *running time* između trenutka odlaska ping poruke, do trenutka primitka pong odgovora.

3 Primjena u nastavi fizike

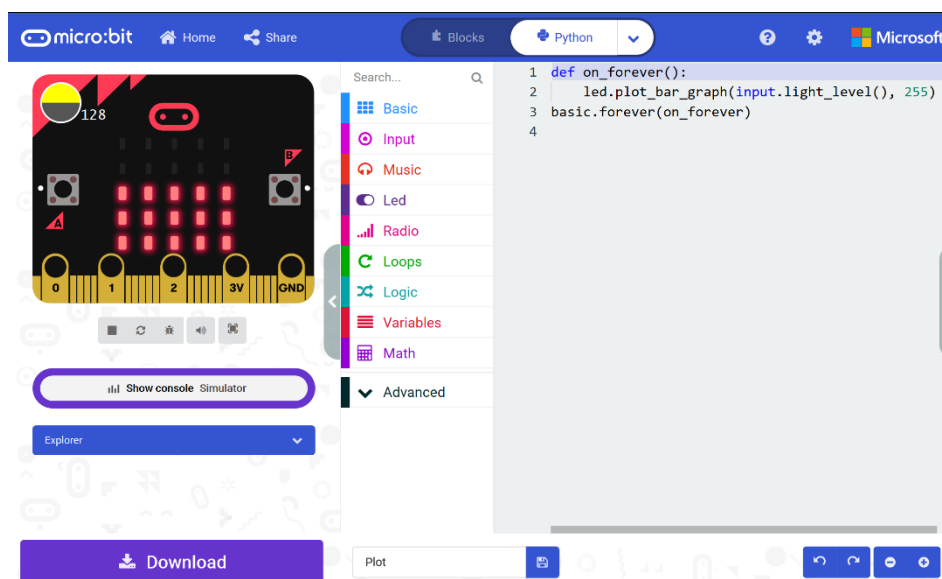
3.1 Prikupljanje podataka upotrebom micro:bit uređaja

Podatke tijekom izvođenja fizikalnih pokusa možemo prikupljati na tri načina. Prvi ispisuje podata na zaslonu micro:bita i njihovo zapisivanje, drugi koristi USB kabel za prijenos podataka na računalo, a treći prijenos podataka na računalo ili mobitel vrši radio vezom, ili kombinacijom radio veze s USB kabelom.

U jednostavnijim slučajevima kada nema puno mjerenja, dovoljno je koristiti LED zaslon uređaja za prikaz podataka, koji se zatim mogu zapisivati na papir. Da bismo npr. sensorima izmjerenu vrijednost temperature prikazali pomoću LED zaslona, potrebno je koristiti funkciju `basic.show_number(input.temperature())`.

Podatke je na zaslonu micro:bita također moguće grafički prikazati koristeći funkciju `led.plot_bar_graph(input.temperature(), max_temp)`, gdje je `max_temp` cijeli broj koji se grafički prikazuje uključivanjem svih 25 LED žaruljica micro:bit uređaja, čime se određuje maksimalna temperatura koju je moguće grafički prikazati LED žaruljicama (Slika 16). Kao primjer, za `max_temp = 50`, temperatura od 10°C čini 20% maksimalne temperature, što će grafički biti prikazano uključivanjem 5 žaruljica donjeg reda, odnosno uključiti će se samo 20% sveukupnog broja žaruljica.

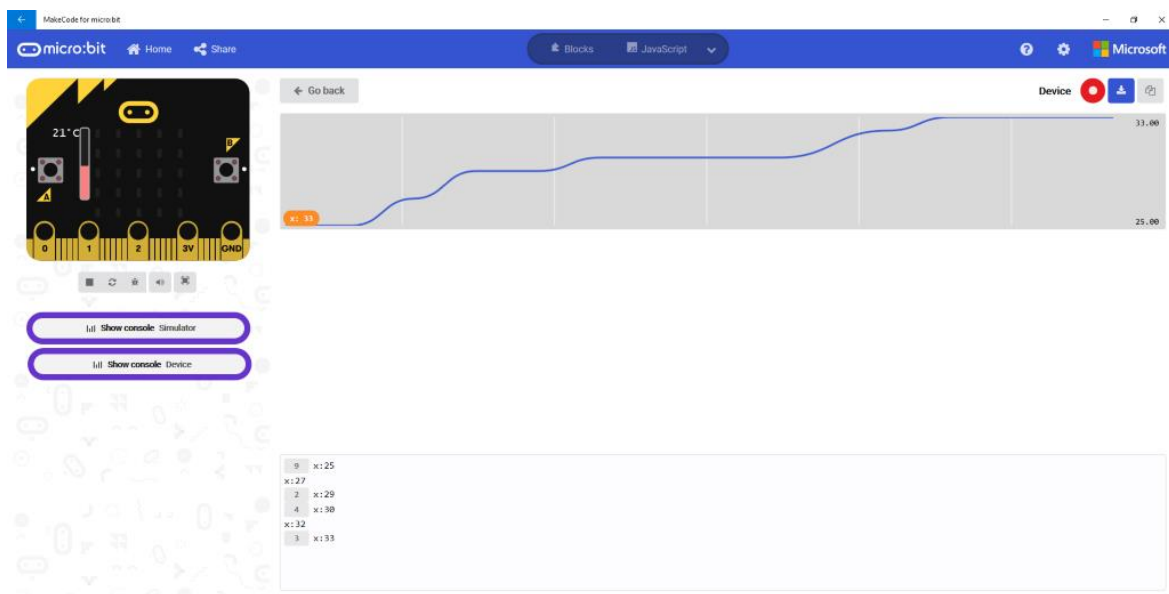
Obzirom da smo ograničeni brojem žaruljica uređaja, ovaj način grafičkog prikazivanja podataka nije dovoljno precizan, ali daje kvalitativne odnose i upoznaje učenike s grafički prikazom izmjerenih varijabli.



Slika 16 Prikaz makecode simulatora i programskog koda koji izmjerenu osvijetljenost grafički prikazuje koristeći LED diode uređaja

Detaljniji grafički prikaz podataka moguće je dobiti spajanjem micro::bit uređaja s računalom putem USB kabela, čime je MakeCode aplikaciji omogućen trenutni pristup mjerenim podacima, uz pomoć već korištene funkcije , ili koristeći npr. `serial.write_value("x", input.temperature())`, u slučaju kada je potrebno bilježiti više fizikalnih veličina koje se mjere različitim sensorima (Slika 18).

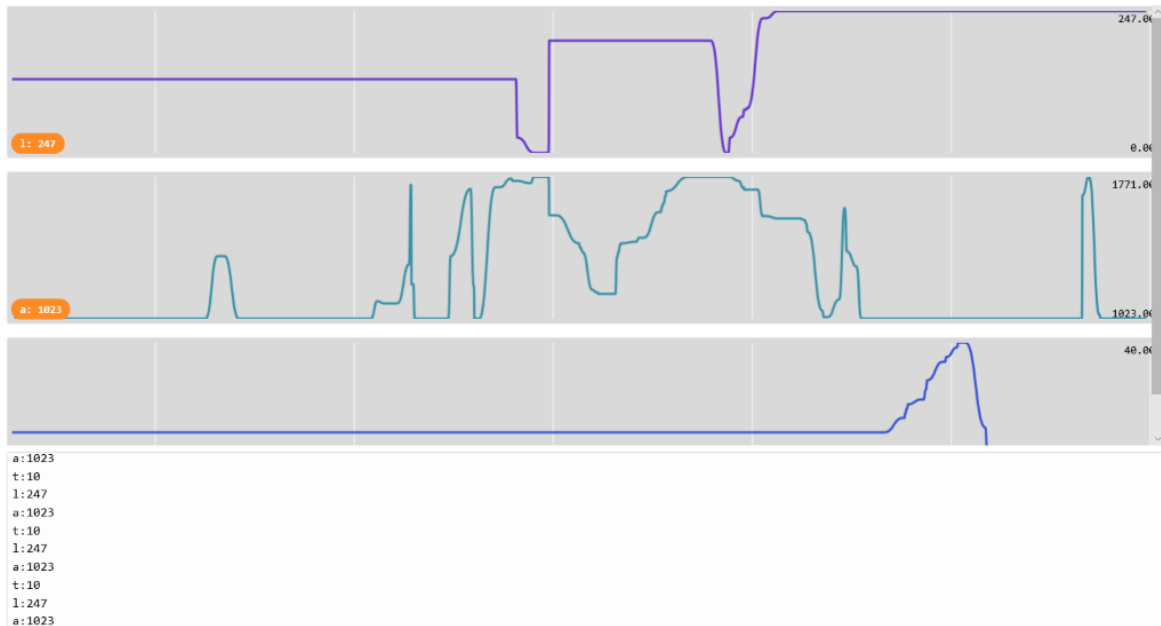
Kada je program pokrenut, ispod simulatora se pojavi tipka *Show console Device*, koja vodi do mjerenih podataka, te grafičkom prikazu istih (Slika 17). Iznad iscrtanih grafova nalaze se tipke za zaustavljanje/pokretanje prikupljanja podataka, preuzimanje prikupljenih podataka u CSV formatu, te kopiranje prikupljenih podataka u .txt formatu, za kasniju analizu.



Slika 17 Prikaz vrijednosti izmjerenih fizikalnih varijabli na računalu.

Primjer programskog koda za istovremeno prikupljanje podataka s različitih senzora

```
def on_forever():  
    serial.write_value("t", input.temperature())  
    serial.write_value("l", input.light_level())  
    serial.write_value("a", input.acceleration(Dimension.STRENGTH))  
  
basic.forever(on_forever)
```



Slika 18 Prikaz očitanih vrijednosti (temperatura, osvjetljenje, ubrzanje)

Obzirom da micro:bit uređaj često prilikom izvođenja demonstracijskih pokusa i raznih mjerenja nije stacionaran, ili na odgovarajućem položaju, javlja se problem povezanosti samog uređaja s računalom putem USB kabela, zbog čega je praktično koristiti dva micro:bit uređaja međusobno povezana *radio vezom*. Pritom se jedan uređaj koristi za mjerenje neke fizikalne veličine i bežično slanje izmjenjenih podataka putem radiovalova drugom uređaju, koji je povezan USB kablom s računalom, za prikaz i pohranu izmjenjenih fizikalnih veličina.

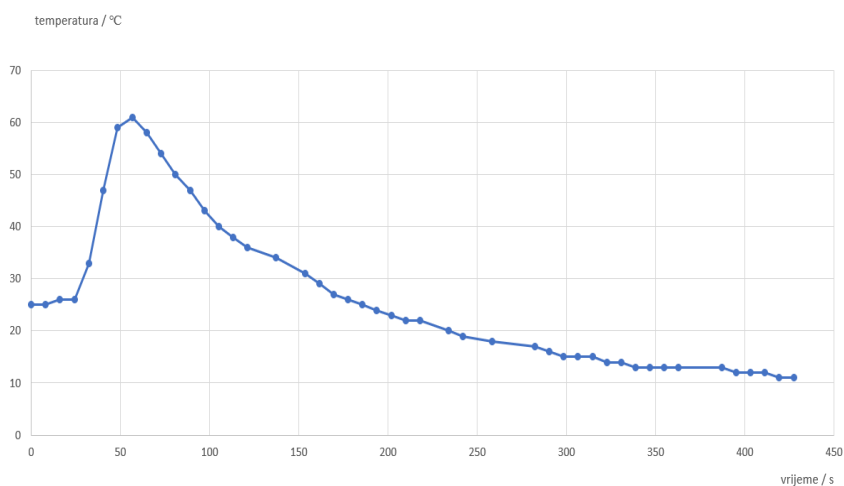
3.2 Mjerenje temperature

Važnu ulogu bežične komunikacije učenici lako spoznaju već pri izvođenju jednostavnijih mjerenja, kao što je mjerenje vanjske temperature. Temperaturni senzor jedan je od prvih mjernih instrumenata koje početnici upoznaju, a kada se upari s radio vezom pomoću koje micro:bit uređaj izmjerenu temperaturu šalje drugom prijemnik-uređaju, mjerenje temperature dobiva novu dimenziju. Povežemo li prijemnik s računalom, moguće je izmjerene vrijednosti prikupljati tijekom duljeg perioda, a zatim ih preuzeti u obliku csv, ili tekstualne datoteke. Dobiveni podaci mogu se dalje obrađivati u okviru nastavnog procesa (Slika 19).

U jednostavnijem slučaju prijemnik temperature, izmjerene i poslane od odašiljača, prikazuje na ekranu, što za opširnije prikupljanje mjerenja i daljnju obradu podataka zahtjeva ručno zapisivanje izmjerenih temperatura.

Automatsko mjerenje i pohranjivanje izmjerenih temperatura, koje ne zahtjeva prisutnost čovjeka pri mjerenju, i ručno bilježenje podataka, moguće je dobiti povezivanjem micro:bit prijemnika s računalom, putem USB kabla. Na taj način sve temperature, koje prijemnik dobije putem radiovalova, ostaju zabilježene, upotrebom `serial.write_value()` funkcije.

1	time (source1)	Celcius
2	0	25
3	8.059	25
4	16.183	26
5	24.241	26
6	32.365	33
7	40.489	47
8	48.548	59
9	56.672	61
10	64.797	58
11	72.854	54
12	80.913	50
13	89.043	47
14	97.094	43
15	105.154	40
16	113.212	38
17	121.27	36
18	137.386	34
19	153.503	31
20	161.56	29



Slika 19 Izmjerene vrijednosti temperature tijekom (relativno) naglog zagrijavanja i hlađenja na temperaturu okoline

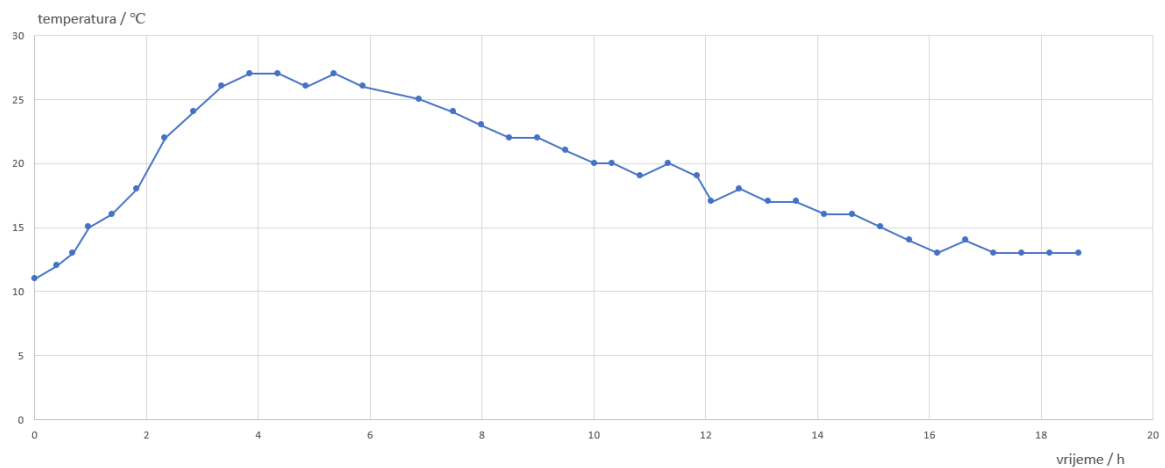
Programski kod za mjerenje temperature - odašiljač

```
temperature = 0
basic.show_string("ODASILJAC")
radio.set_group(99)

def on_forever():
    global temperature
    temperatura = input.temperature()
    basic.show_number(temperatura)
    radio.send_number(temperatura)
    basic.pause(1000)
basic.forever(on_forever)
```

Prilikom pokretanja programa na micro:bit uređaju prikazuje se znakovni niz 'Odašiljač', postavlja se radio grupa i vrijednost varijable *temperatura* = 0. U petlji *forever* se sensorima izmjerena vrijednost temperature pridružuje varijabli *temp*, prikazuje brojčano pomoću LED zaslona i odašilje putem radio signala. Vrijeme između pojedinih mjerenja moguće je odrediti funkcijom *basic.pause()*.

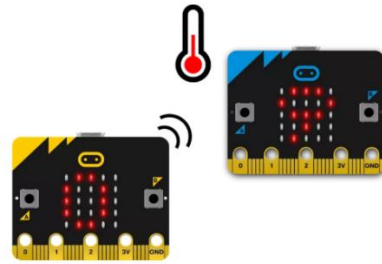
Kod brzih promjena temperature poželjno je koristiti kraće vremenske intervale, dok u slučaju sporijih promjena, na primjer praćenja temperature tijekom cijelog dana, dovoljno je mjerenja vršiti u većim vremenskim razmacima (Slika 20). Da bismo takva mjerenja učinili što preciznijim, moguće je unaprijediti programski kod kojim bi micro:bit u ulozi odašiljača uspoređivao trenutnu temperaturu s prethodno izmjerenom (u kraćim vremenskim intervalima), te samo u slučaju promjene temperature slao novo izmjerenu vrijednost temperature.



Slika 20 Grafički prikaz cjelodnevnog mjerenja i praćenja temperature

Programski kod – prijemnik prikazuje temperaturu LED žaruljicama

```
temp = 0
radio.set_group(99)
basic.show_string("PRIJEMNIK")
def on_received_number(receivedNumber):
    basic.show_number(receivedNumber)
radio.on_received_number(on_received_number)
```



Kao i kod programskog koda namijenjenog slanju podataka, poželjno je prilikom pokretanja prijemnika prikazati znakovni niz koji opisuje ulogu micro:bit uređaja (odašiljač ili prijemnik). Također je potrebno namjestiti radio grupu na istu vrijednost koja je postavljena u programskom kodu odašiljača, te odrediti početnu vrijednost pohranjenu u globalnoj varijabli *temp*.

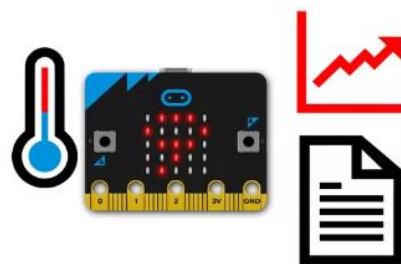
Koristeći funkciju *on_received_number()* micro:bit prijemnik registrira temperaturu poslanu od strane odašiljača, nakon čega je prikazuje pomoću LED žaruljica. Proces se ponavlja svaki put kada prijemnik primi radio signal odašiljača.

Programski kod – prijemnik prosljeđuje podatke računalu

Dodatnim redom koda bežično zaprimljen broj moguće je pridružiti nizu "*t(Celsius)*" i prosljediti MakeCode aplikaciji putem USB veze.

```
temperature = 0
basic.show_string("PRIJEMNIK SERIAL")
radio.set_group(99)
def on_received_number(receivedNumber):
    basic.show_number(receivedNumber)

serial.write_value("Celsius", receivedNumber)
radio.on_received_number(on_received_number)
```

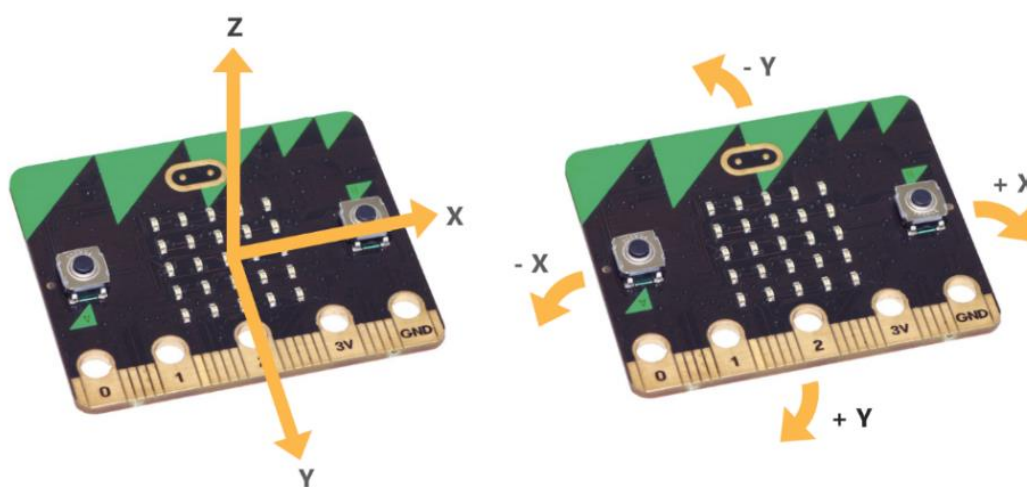


Škole koje koriste veći broj micro:bit uređaja, mogu ih koristiti kao meteorološku stanicu, a za prikaz trenutne vanjske i unutarnje temperature dovoljno je upotrijebiti jedan micro:bit prijemnik koji bi ispisivao sve podatke svojim LED žaruljama.

3.3 Mjerenje ubrzanja i slanje podataka putem radio veze

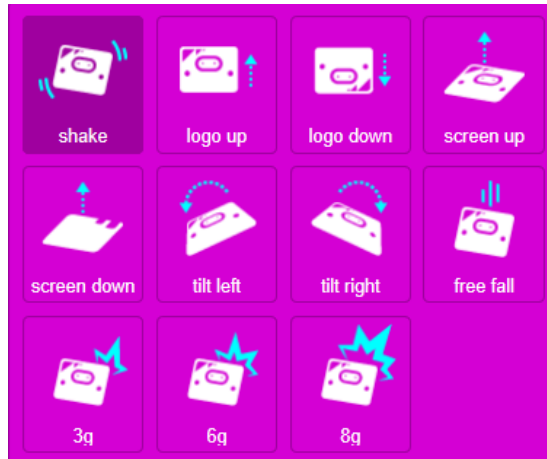
Akcelerometar, ili mjerač ubrzanja je mjerni instrument ili uređaj za mjerenje akceleracije (ubrzanja) tijela u navigaciji, aeronautici, seizmologiji. Upotrebljava se i za ispitivanje vibracija, udaraca i slične svrhe.

Micro:bit ima *akcelerometar* koji mjeri iznos i smjer ubrzanja koje je superpozicija ubrzanja sile teže i ubrzanja samog uređaja (ako uređaj miruje, izmjereno ubrzanje jednako je ubrzanju sile teže). Na temelju promjene ubrzanja uređaj može detektirati niz standardnih radnji kao što su trešnja, nagib i slobodni pad. Ubrzanje možemo promatrati u Kartezijevom koordinatnom sustavu gdje su osi naznačene kao na Slika 21.



Slika 21 Koordinatni sustav micro:bita

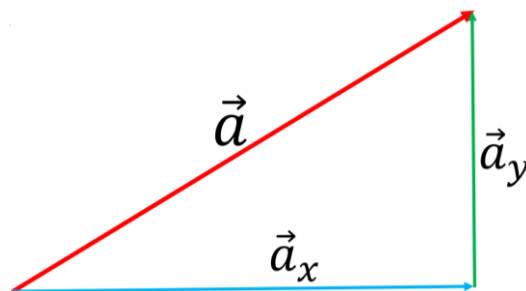
Vrijednost ubrzanja duž svake osi je pozitivan ili negativan broj čija vrijednost je u mili-g mjernoj jedinici ($10^{-3} \text{ a} / \text{g}$), gdje je g ubrzanje sile teže $g=9,81\text{m/s}^2$. Na primjer polegnemo li micro:bit na horizontalnu podlogu, tako da je prednja strana okrenuta gore, vrijednosti ubrzanja za X i Y smjer približno su zanemarive, dok vrijednost u Z mjeru iznosi -1024, što govori da ubrzanje na uređaj u takvom položaju, djeluje prema dolje, odnosno u -Z smjeru. Zarotiramo li micro:bit na suprotnu stranu, tako da LED lampice gledaju prema dolje, vrijednost ubrzanja u Z smjeru iznositi će +1024. Koristeći akcelerometar, micro:bit prepoznaje različite geste odnosno pokrete (Slika 22).



Slika 22 Geste koje micro:bit prepoznaje na temelju mjerenja akcelerometra.

Za ispravnu interpretaciju gesta, ubrzanje je potrebno promatrati kao vektor u trodimenzionalnom prostoru. Ubrzanje je vektorska veličina, definirana iznosom i smjerom. Npr. za računanje iznosa ubrzanja u x-y ravnini (Slika 23) potrebno je koristiti Pitagorin poučak:

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2}$$



Slika 23 Vektorski prikaz akceleracije s komponentama u 2D prostoru

Isti princip vrijedi i za 3D akcelerometre, kod kojih ukupna sila ovisi i o Z komponenti, prema izrazu:

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Mjerimo li ukupan iznos ubrzanja, za uređaj koji miruje ono će iznositi $1g$, neovisno o orijentaciji/usmjerenju micro:bit uređaja. Očekivano je da čak i za vrijeme mirovanja vrijednost mjerena akcelerometrom fluktuiraju, kao posljedica nesavršenosti uređaja. Proces kalibracije senzora često pomaže kada se ukaže potreba za preciznim određivanjem mjerenih veličina.

Primjer programskog koda za mjerenje akceleracije – odašiljač

Pokretanjem micro:bit uređaja na ekranu se ispisuje znakovni niz „GRAVITY SENDER“. Potrebno je postaviti istu radio grupu koju će koristiti primatelj. Petlja forever konstantno promatra iznos ubrzanja, čiju vrijednost šalje radio signalom uređajima s istom adresom grupe.

```
basic.show_string("GRAVITY SENDER")
radio.set_group(99)
def on_forever():
    radio.send_number(input.acceleration(Dimension.STRENGTH))
basic.forever(on_forever)
```

Primjer programskog koda za mjerenje akceleracije – prijemnik

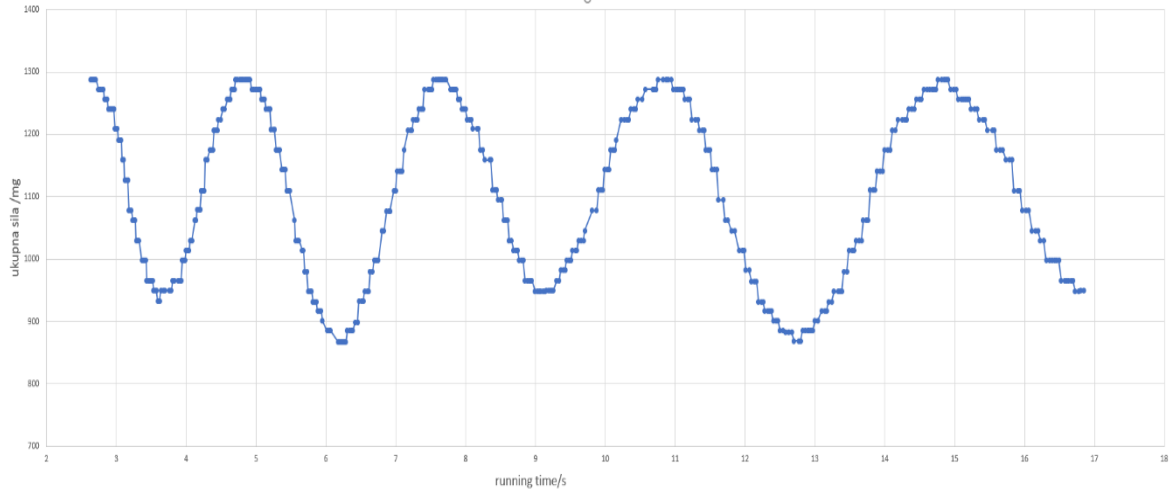
```
basic.show_string("GRAVITY RECEIVER")
radio.set_group(99)
def on_received_number(receivedNumber):
    serial.write_value("gravity", receivedNumber)
radio.on_received_number(on_received_number)
```

Micro:bit koji se koristi kao prijemnik ostaje povezan s računalom putem USB kabela, za prikaz primljenih podataka. Pokretanjem prijemnika ispisuje se pripadni znakovni niz kojim ga razlikujemo od odašiljača. Funkcija *on_received_number()* neprekidno nadgleda radio signale svoje grupe. Vrijednost zaprimljena radio vezom pridružuje se varijabli „gravity“.



Slika 24 Promatranje zabilježene akceleracije tijekom titranja matematičkog njihala

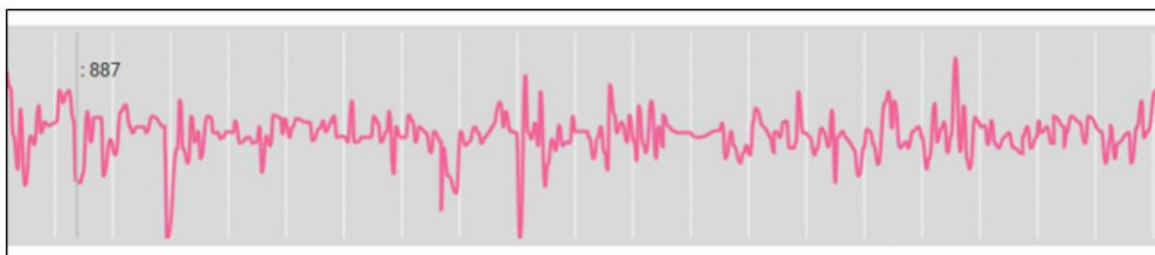
Izmjerene, poslane i zaprimljene podatke moguće je grafički promatrati unutar MakeCode aplikacije (Slika 24), ili ih pohraniti u .csv formatu, i zatim koristiti za obradu ili grafički prikaz (Slika 25).



Slika 25 Grafički prikaz akceleracije u ovisnosti o vremenu za matematičko njihalo

3.3.1 Senzor zvučnih valova, detektor potresa

Akcelerometar micro:bit uređaja može se koristiti i za detektiranje zvuka, odnosno zvučnih vibracija (Lyman, 2018). Potrebno je uređaj, koji služi kao „mikrofon“, postaviti blizu nekog izvora zvuka (npr. zvučnika), kako bi izazvali što veća titranja. Zabilježene zvučne valove potrebno je poslati radio vezom prijemniku, koji omogućuje da zaprimljene vrijednosti proučavamo grafički (Slika 26).



Slika 26 Detektirani zvučni valovi

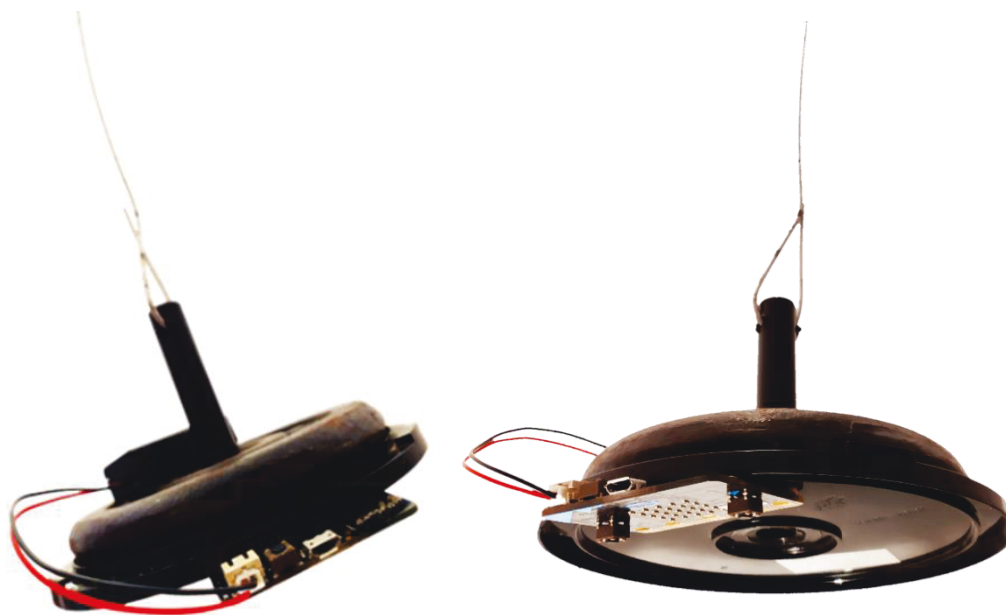
Potresi uzrokuju vibracije zemljinog tla, koje mogu bit također detektirane pomoću akceleratora ugrađenog u micro:bit uređaj (Slika 27), koristeći isti program kao u prošlim primjerima, kod „snimanja“ zvuka i mjerenje akceleracije općenito. Postavimo li micro:bit na mirnu podlogu i namjestimo li manji mjerni opseg uređaja, čime povećamo osjetljivost akcelerometra, moguće je detektirati i zabilježiti vibracije zemljinog tla, pa samim time koristiti micro:bit uređaj za uključivanje alarma (Lyman, 2018).



Slika 27 Detektiranje trešnje (potresa)

3.3.2 Matematičko njihalo

Matematičko je njihalo najjednostavnije njihalo čiji period ne ovisi o masi već samo o ubrzanju sile teže i duljini niti. Jedan micro:bit mjeri period u milisekundama, a drugi s tim podatkom izračuna frekvenciju i duljinu niti (Edutorij, n.d.). Za projekt su potrebna 2 micro:bita, njihalo (stalak i nit), oprema za pričvršćivanje uređaja za njihalo (ljepljiva traka, elastična guma i sl.), te izvor svjetlosti(lampa).



Slika 28 Matematičko njihalo

Sastavite/postavite stalak za njihalo, izradite nit njihala tako da njihalo bude što stabilnije. Nit objesite na stalak za kojeg se može po potrebi pričvrstiti “drugi” micro:bit, koji služi kao prijemnik. Za njihalo prikazano na slikama korištena je kutija za CD medije, obješena o nit njihala (Slika 28). Umjesto CD medija na stalak je postavljen uteg mase 1kg, koji osigurava da većina mase njihala bude koncentrirana uz sam rub niti, čime masa niti postaje zanemariva. Na njihalo je potrebno pričvrstiti držač za baterije i drugi micro:bit uređaj, koji služi za mjerenje perioda.

Lampu, odnosno izvor svjetlosti centrirajte u ravnotežni položaj njihala (položaj u kojem se nalazi micro:bit kad se ne njiše), tako da osvjetljava LED diode micro:bit uređaja (Slika 29). Najbolje je koristiti usmjereni snop svjetlosti ili laser.



Slika 29 Postav pokusa s matematičkim njihalom

Program za prvi micro:bit

Za potrebe radio komunikacije postavljamo radio grupu (u primjeru je odabrana vrijednost 7), a varijablu svjetlo na očitane vrijednosti *light level*, kako bi se micro:bit kalibrirao, odnosno očitao uobičajenu jakost svjetlosti.

```
radio.set_group(7)
svjetlo = input.light_level()
```

Kako bi, za vrijeme dok je gumb A nepritisnut, micro:bit mirovao, a pritiskom i otpuštanjem gumba A započeo vršiti mjerenje, koristi se petlja while:

```
while not (input.button_is_pressed(Button.A)) :
    basic.pause(50)
```

Otpuštanjem gumba A na led lampicama se ispisuje znakovni niz "Idi", kao znak da je potrebno zanjhati micro:bit. Dio koda također postavlja potrebne varijable na njihovu početnu vrijednost.

```
basic.show_string("Idi")
svjetlo = input.light_level()
brojac = 0
brpoltitraja = 18
titraji = Math.idiv(brpoltitraja, 2)
trensvjetlo = 0
t1 = 0
```

Za preciznije izvođenje mjerenja, potrebno je izmjeriti vrijeme potrebno za veći broj titraja matematičkog njihala, a zatim, iz izmjenjenog vremena izračunati vrijeme/period jednog titraja. Mjerenje određenog broja titraja moguće je realizirati koristeći while petlju, koja obavlja zadatak toliko dugo dok je varijabla *brojac* manja ili jednaka varijabli *brpoltitraja* te postavlja varijablu *trensvjetlo* na trenutnu jačinu svjetlosti/light level.

```
while brojac <= brpoltitraja:
    trensvjetlo = input.light_level()
```

If uvjet uspoređuje vrijednost varijable *trensvjetlo* s početno izmjenjenom osvjetljenosti, koja je pohranjena u varijabli *svjetlo*. Ukoliko je vrijednost varijable *trensvjetlo* veća za od varijable *svjetlo*, za minimalni iznos od 50, na ekranu će se prikazati znak + (blok show leds), koji se nakon 50 ms isključi, a varijabli *brojac* poveća vrijednost za 1. Ako se to dogodi po prvi put, micro:bit započinje brojati vrijeme, pridružujući vrijednost running time varijabli *t1* (if naredba čiji je uvjet da je *t1* jednak nuli).

```
if trensvjetlo > svjetlo + 50:
    if t1 == 0:
        t1 = input.running_time()
        basic.show_leds("""
            . . # . .
            . . # . .
            # # # # #
            . . # . .
            . . # . . """)
        basic.pause(50)
        basic.clear_screen()
        brojac += 1
```

Kada brojač dostigne željenu vrijednost (izmjerimo željeni broj titraja), varijabli *t2* pridružuje se vrijednost *running time*, čime je omogućen izračuna perioda, odnosno varijable *T*, tako da razliku *t2* i *t1* podijelimo s brojem titraja.

```
t2 = input.running_time()
T = Math.idiv(t2 - t1, titraji)
```

Koristeći funkciju *show_number()* program prikazuje izračunatu vrijednost varijable *T*.

```
basic.show_string("::: T=")
basic.show_number(T)
```

Upotrebom *on_button_pressed_()* i *radio_send_number()* omogućuje se slanje broja *T* na drugi micro:bit.

```
def on_button_pressed_b():
    radio.send_number(T)
input.on_button_pressed(Button.B, on_button_pressed_b)
```


Konačan programski kod:

```
T = 0
svjetlo = 0
brojac = 0
brpoltitraja = 0
titraji = 0
trensvjetlo = 0
t1 = 0
t2 = 0

def on_button_pressed_b():
    radio.send_number(T)
input.on_button_pressed(Button.B, on_button_pressed_b)

def on_forever():
    global svjetlo, brojac, brpoltitraja, titraji, trensvjetlo, t1
    , t2, T
    radio.set_group(7)
    svjetlo = input.light_level()
    while not (input.button_is_pressed(Button.A)):
        basic.pause(50)
    basic.show_string("Idi")
    svjetlo = input.light_level()
    brojac = 0
    brpoltitraja = 18
    titraji = Math.idiv(brpoltitraja, 2)
    trensvjetlo = 0
    t1 = 0
    while brojac <= brpoltitraja:
        trensvjetlo = input.light_level()
        if trensvjetlo > svjetlo + 50:
            if t1 == 0:
                t1 = input.running_time()
                basic.show_leds("""
                    . . # . .
                    . . # . .
                    # # # # #
                    . . # . .
                    . . # . . """)
                basic.pause(50)
                basic.clear_screen()
                brojac += 1
            t2 = input.running_time()
            T = Math.idiv(t2 - t1, titraji)
            basic.show_string("::: T=")
            basic.show_number(T)
    basic.forever(on_forever)
```

Program za drugi microbit

Micro:bit pričvršćen za stalak njihala, za prijem izmjerenih vrijednosti zahtjeva postavljenje iste vrijednosti radio grupe kakva je postavljena u programskom kodu prvog micro:bit uređaja (odašiljača). Radio vezom zaprimljeni broj sprema se u varijablu T, i prikazuje na ekranu.

```
def on_received_number_deprecated(receivedNumber):
    global T
    T = receivedNumber
    basic.show_number(receivedNumber)
```

Varijabla item (frekvencija) biti će izračunata i prikazana na ekranu pritiskom na gumb A. Frekvencija se inače računa izrazom $1/T$, ali kako micro:bit ne radi s decimalnim brojevima brojevima, i T je u ms, a ne sekundama, broj 1 uvećan je 100000 puta da bi se dobile prve dvije decimale frekvencije. (Burk, n.d.)

```
def on_button_pressed_a():
    global item
    item = Math.idiv(100000, T)
    basic.show_string("0," + str(item))
input.on_button_pressed(Button.A, on_button_pressed_a)
```

Za izračunavanje duljine niti potrebno je postaviti varijable kojima pridružujemo vrijednosti konstanti.

```
def on_button_pressed_b():
    global dvapi, g, l
    dvapi = 314 * 3141 * 4
    g = 98
    l = Math.idiv(g * T * T, dvapi)
    basic.show_number(l)
input.on_button_pressed(Button.B, on_button_pressed_b)
```

Unesene formule za izračun varijable l , odnosno duljine niti, potrebno je pokrenuti pritiskom na gumb B, koja će ujedno prikazati izračunatu vrijednost, koristeći `show_number(l)`.

Konačan kod:

```

T = 0
item = 0
dvapi = 0
g = 0
l = 0
def on_received_number_deprecated(receivedNumber):
    global T
    T = receivedNumber
    basic.show_number(receivedNumber)
radio.on_received_number_deprecated(on_received_number_deprecated)
def on_button_pressed_a():
    global item
    item = Math.idiv(100000, T)
    basic.show_string("0," + str(item))
input.on_button_pressed(Button.A, on_button_pressed_a)
def on_button_pressed_b():
    global dvapi, g, l
    dvapi = 314 * 3141 * 4
    g = 98
    l = Math.idiv(g * T * T, dvapi)
    basic.show_number(l)
input.on_button_pressed(Button.B, on_button_pressed_b)
def on_forever():
    radio.set_group(7)
basic.forever(on_forever)

```

Rezultati mjerenja

l=71cm	period / ms	period / s	g	l / m
	1750	1,75	9,1432699	0,76177342
	1740	1,74	9,2486669	0,75309232
	1734	1,734	9,3127823	0,74790753
	1748	1,748	9,1642047	0,76003322
	1748	1,748	9,1642047	0,76003322
l=135cm	2363	2,363	9,5351079	1,38891978
	2380	2,38	9,3993786	1,40897612
	2380	2,38	9,3993786	1,40897612
	2363	2,363	9,5351079	1,38891978
	2363	2,363	9,5351079	1,38891978
	2369	2,369	9,4868697	1,39598207
	2369	2,369	9,4868697	1,39598207
l=100cm	2058	2,058	9,3116908	1,05351436
	2044	2,044	9,4396851	1,03922959
	2042	2,042	9,4581852	1,03719686
	2058	2,058	9,3116908	1,05351436
	2047	2,047	9,4120365	1,0422824

Tablica 4

3.3.3 *Micro:bit alarmni sustav*

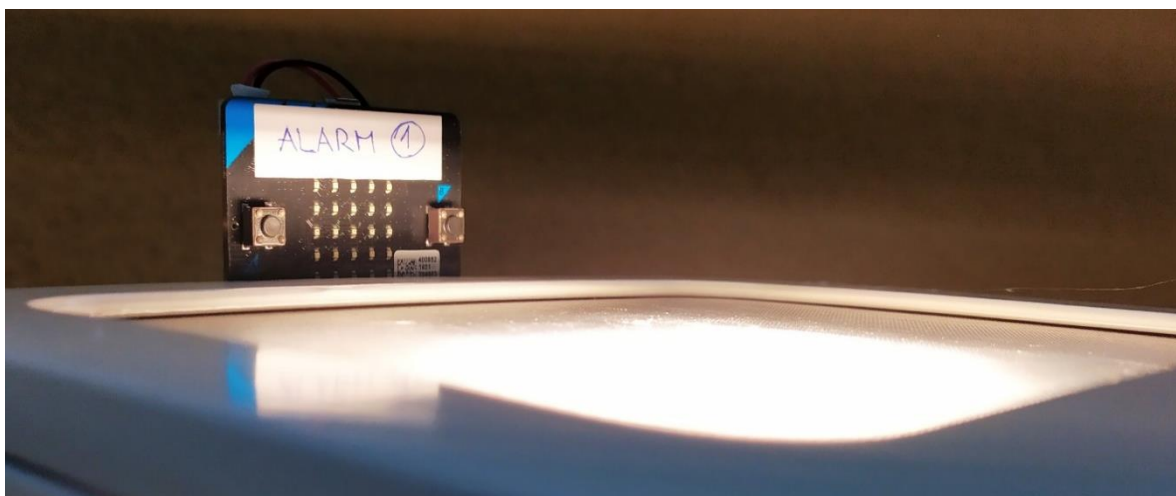
Micro:bit često se koristi za razne projektne zadatke, pa je tako sa većim brojem micro:bit uređaja moguće demonstrirati izradu alarmnih sustava, upotrebom radio komunikacije i raznih senzora za detekciju.

Tri micro:bita služe detekciji četiri neželjenih pojava – požar, otvaranje vrata, uključivanje svjetla, i kretanje osobe osvijetljenom prostorijom. Četvrti, alarmni micro:bit reagira na radio signale poslane od prva tri micro:bita razlikujući vrste neželjenih pojava.

Priprema opreme

Prvi micro:bit služi detekciji uključivanja svjetla. Ovaj je micro:bit najbolje postaviti blizu glavne lampe u kući (npr. u hodniku ili dnevnoj sobi) ukoliko osoba koja je ušla u sobu uključi svjetlo, jer će tada najbolje izmjeriti promjenu jačine svjetlosti, iako će reagirati i na ručne svjetiljke (Slika 30).

Optimalno rješenje je postavljanje više micro:bitova na nekoliko mjesta u kući. Najvažnije je kod postavljanja micro:bita ne prekriti LED žaruljice te ih usmjeriti prema ulaznim vratima kroz koje bi neželjena osoba mogla ući, odnosno usmjeriti ih u smjeru s koje bi svjetlost najvjerojatnije mogla doći.

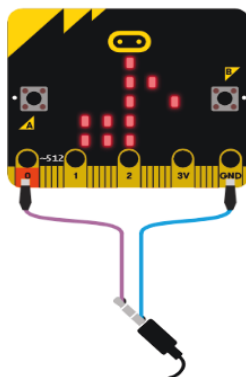


Slika 30 Micro:bit namijenjen za detekciju uključivanja svjetla

Drugi micro:bit ima dvije svrhe – služi detekciji otvaranja ulaznih (ili drugih) vrata i detekciji požara. Potrebno ga je pričvrstiti za vrata – npr. nekom vrstom ljepljive trake (selotejp, patafix...) ili nekim stalkom koji se pričvrsti na vrata.

Treći micro:bit služi detekciji sjene koju u prolazu baca neželjena osoba. Ovaj je micro:bit vezan za prvi micro:bit i radit će samo ako je uključeno svjetlo (koje detektira prvi micro:bit). To je dodatna provjera miče li se osoba kućom. Ovaj se micro:bit postavlja na suprotnu stranu od lampe kako bi se mogla detektirati sjena koja je bačena na njega.

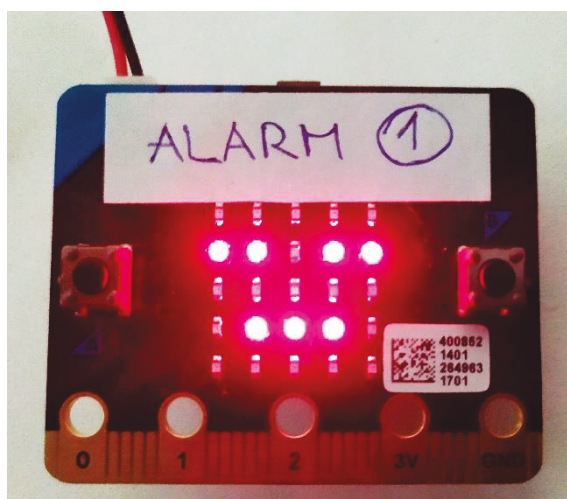
Četvrti micro:bit ima svrhu alarma, odnosno služi dojaviti neželjene pojave. Ovaj je micro:bit najbolje držati čim bliže sebi, npr. u spavaćoj sobi pokraj kreveta. Četvrti se micro:bit mora spojiti na zvučnike jer proizvodi zvukove alarma (Slika 31). Microbit se na zvučnik spaja krokodilkama (Slika 31), a ukoliko ih nemate, spojiti ga možete i alufolijom. Zvučnike treba uključiti i pojačati. Detaljniji opis povezivanja sa zvučnikom/slušalicama nalazi se na kraju poglavlja.



Slika 31 Prikaz povezivanja zvučnika u Makecode simulatoru

Programski kod za prvi micro:bit

Kako bi sva četiri micro:bita bila povezana, koristit će se radiokomunikacija i prvo se za radijsku grupu odabire broj, u ovom slučaju broj 5. Izradimo novu varijablu (*Svjetlost*) i pridružujemo joj vrijednost trenutnog svjetla (*light level*). Ako je izmjerena jačina svjetlosti niža ili jednaka vrijednosti 15, micro:bit je u mraku te se prikazuje slika uspavanog lica kao indikator na samom micro:bitu (Slika 32).



Slika 32 Prikaz uključenih žaruljica kada se micro:bit 1 (detektor svjetla) nalazi u mraku

U suprotnom, ukoliko se uključilo svjetlo, prikazuje se slika iznenađenog lica i preko radiokomunikacije se pošalje broj 1 (kojeg će primiti četvrti, alarmni microbit).

```

Svjetlost = 0
radio.set_group(5)

def on_forever():
    global Svjetlost
    Svjetlost = input.light_level()
    if Svjetlost <= 15:
        basic.show_icon(IconNames.ASLEEP)
    else:
        basic.show_icon(IconNames.SURPRISED)
        radio.send_number(1)
basic.forever(on_forever)

```

Program za drugi micro:bit

Drugom micro:bit uređaju također je potrebno postaviti broj radijske grupe, na istu vrijednost kao i za prvi micro:bit.

U beskonačnu petlju (blok forever) prvo se dodaje provjera (blok if) je li vrijednost akceleracije micro:bita veća od nule, odnosno miče li se micro:bit (otvaraju li se vrata). Ukoliko je vrijednost akceleracije veća od nula šalje se broj 2 preko radiokomunikacije (kojeg će primiti četvrti, alarmni micro:bit).

Isti micro:bit ima dodatnu funkciju – detekcija požara. Dodajemo ponovo beskonačnu petlju (blok forever), te izrađujemo varijablu (*Degree*) u koju će se spremiti trenutna temperatura micro:bita. Zatim micro:bit pauziramo na 3 sekunde (3000 ms) i provjeravamo je li trenutna temperatura veća od temperature izmjerene prije 3 sekunde.

Zbog mogućeg porasta temperature uslijed prirodnih uvjeta (toplina izvana), dodaje se 1 kao tolerirani porast temperature. Ukoliko je temperatura porasla za dva ili više stupnja unutar samo 3 sekunde, šalje se broj 3 preko radiokomunikacije (kojeg će primiti četvrti, alarmni micro:bit) kao znak da je izbio požar.

```

Degree = 0
radio.set_group(5)

def on_forever():
    if input.acceleration(Dimension.Y) > 0:
        radio.send_number(2)
basic.forever(on_forever)

def on_forever2():
    global Degree
    Degree = input.temperature()
    basic.pause(3000)
    if input.temperature() > Degree + 1:
        radio.send_number(3)
basic.forever(on_forever2)

```

Program za treći micro:bit

Treći micro:bit s detekcijom sjene kreće tek kada mu prvi micro:bit pošalje broj 1, odnosno dok svijetli svijetlo u kući što je nužno za detektiranje sjene.

Za provjeru je li, putem radiokomunikacije dohvaćen broj, spremljen u varijablu *receivedNumber*, jednak broju 1, koristi se if blok. Potrebno je dodati i novu varijablu (*Sjena*). Ukoliko je primljen broj 1, prvo se u varijablu *Sjena* sprema vrijednost izmjerene razine svjetlosti (light level).

Dodatni blok if provjerava je li vrijednost izmjerene svjetlosti, tj. vrijednost spremljena u varijabli *Sjena*, niža od 14. Ukoliko jest, šalje se broj 4 preko radiokomunikacije (kojeg će primiti četvrti, alarmni micro:bit) kao znak da je detektirana sjena. Kao indikator na samom trećem micro:bitu, ispisuje se riječ *SJENA*.

```
def on_received_number_deprecated(receivedNumber):
    global Sjena
    if receivedNumber == 1:
        Sjena = input.light_level()
        if Sjena < 14:
            basic.show_string("SJENA")
            radio.send_number(4)
radio.on_received_number_deprecated(on_received_number_deprecated)
Sjena = 0
radio.set_group(5)
```

Program za četvrti (dojavni) micro:bit

Četvrti micro:bit mora reagirati na, preko radiokomunikacije, primljene brojeve 2, 3 i 4. U prvi *else if* blok dodaje se provjera je li vrijednost varijable *receivedNumber* jednaka broju 2, zatim u drugi *else if* blok je li jednaka broju 3. U *else* bloku izvršit će se alarm za primljeni broj 4, što je jedina preostala mogućnost.

U prethodnom koraku izrađeni blok koda za proizvodnju glazbe i žmirkanja LED žaruljica potrebno je duplicirati, te dodati u preostala 3 ishoda grananja (u dva *else if* bloka i *else* blok), uz izmijenjenu vrstu melodije (u primjeru je redom prelude, ringtone i funk).

```
def on_received_number_deprecated(receivedNumber):
    if receivedNumber == 1:
        music.start_melody(music.built_in_melody(Melodies.CHASE,
        MelodyOptions.FOREVER)
        while True:
            led.set_brightness(255)
            basic.show_leds("""
                # # # # #
                # # # # #
                # # # # #
                # # # # #
                # # # # #""")
            basic.pause(50)
            led.set_brightness(0)
            basic.show_leds("""
                # # # # #
                # # # # #
                # # # # #
                # # # # #
                # # # # #""")
    elif receivedNumber == 2:
        music.start_melody(music.built_in_melody(Melodies.PRELUDE)
        , MelodyOptions.FOREVER)
        while True:
            led.set_brightness(255)
            basic.show_leds("""
                # # # # #
                # # # # #
                # # # # #
                # # # # #
                # # # # #""")
            basic.pause(50)
            led.set_brightness(0)
            basic.show_leds("""
                # # # # #
                # # # # #
                # # # # #
                # # # # #
                # # # # #""")
    elif receivedNumber == 3:
        music.start_melody(music.built_in_melody(Melodies.RINGTONE),
        MelodyOptions.FOREVER)
```



```

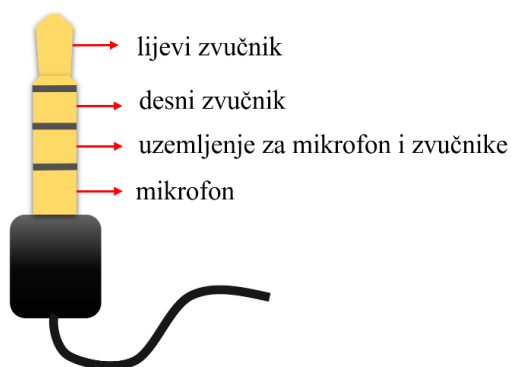
while True:
    led.set_brightness(255)
    basic.show_leds("""
        # # # # #
        # # # # #
        # # # # #
        # # # # #
        # # # # #""")
    basic.pause(50)
    led.set_brightness(0)
    basic.show_leds("""
        # # # # #
        # # # # #
        # # # # #
        # # # # #
        # # # # #""")
else:
    music.start_melody(music.built_in_melody(Melodies.FUNK),
        MelodyOptions.FOREVER)
    while True:
        led.set_brightness(255)
        basic.show_leds("""
            # # # # #
            # # # # #
            # # # # #
            # # # # #
            # # # # #""")
        basic.pause(50)
        led.set_brightness(0)
        basic.show_leds("""
            # # # # #
            # # # # #
            # # # # #
            # # # # #
            # # # # #""")
radio.on_received_number_deprecated(on_received_number_deprecated)
radio.set_group(5)

```

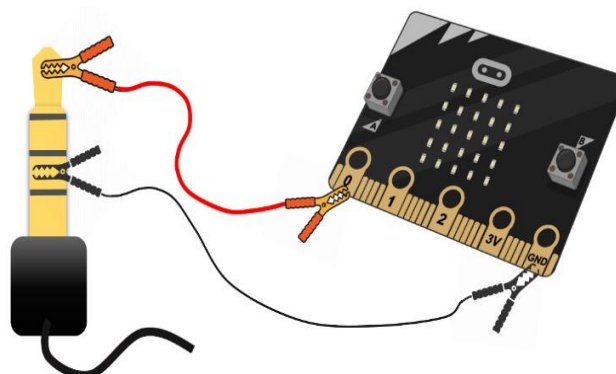
Povezivanje micro:bit uređaja sa zvučnikom

Za reprodukciju zvuka pomoću micro:bit uređaja potrebno ga je, preko izvoda, povezati s izvorom zvuka (slušalicama, zvučnicima za računalo, autozvučnicima itd.). Kao što je već prikazano na Slika 3, microbit PIN priključci namijenjeni su komunikaciji s vanjskim uređajima ili napajanju. Velikih pet priključaka označeni su redom *P0*, *P1*, *P2*, *3V* i *GND*, na koje je moguće povezati vanjske uređaje tzv. „krokodilkama“. *P0*, *P1* i *P2* služe za spajanje micro:bit uređaja sa različitim sensorima i dodacima (npr. elektromotor, pumpa, senzor vlage, senzor razine vode, zvučnik...). *3V* i *GND* pinovi služe za napajanje vanjskih uređaja.

Za povezivanje zvučnika putem velikih izvoda, krajeve jednog krokodil kabela potrebno je učvrstiti na izvod *GND* micro:bita i uzemljeni dio 3,5 milimetarskog priključka slušalica (ukoliko slušalice nisu opremljene mikrofonom, izvod uzemljenja nalazi se uz bazu 3,5mm priključka). Krajevi drugog kabela povezuju naponski izvod *P0* micro:bit uređaja s pozitivnim naponskim dijelom (+) na priključku zvučnika ili slušalica.

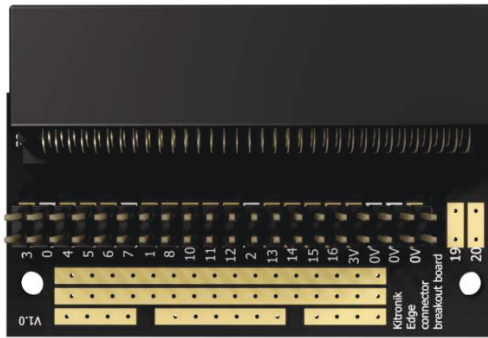


Slika 33



Slika 34

Za razliku od velikih izvoda (pinova), na koje se izravno mogu spajati dodatne komponente (senzori, razni izlazni uređaji), male pinove potrebno je priključiti putem EDGE konektora (Slika 35), na kojeg se spajaju ostali uređaji. Povezivanje zvučnika putem EDGE konektora ne razlikuje se mnogo u odnosu na priključivanje putem velikih izvoda, potrebno je odgovarajuće izvode EDGE konektora povezati na isti način s priključcima zvučnika slušalica.

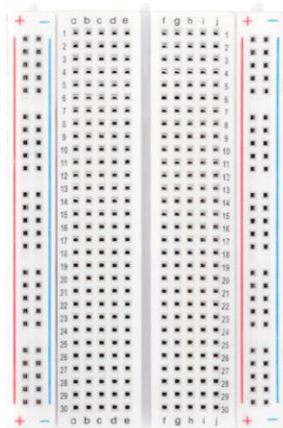


Slika 35 EDGE konektor

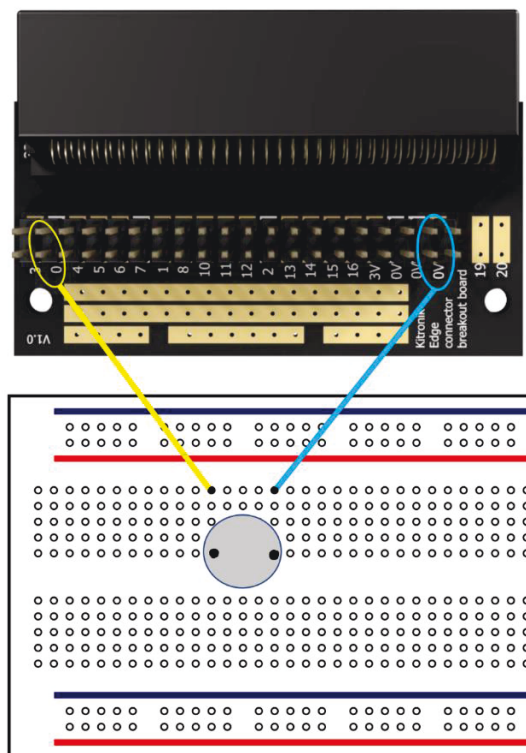
Obzirom da EDGE konektor dolazi u sklopu Inventor's kit paketa, koji sadrži ploču za povezivanje (Slika 38) i piezoelektrični zvučnik (Slika 36), odnosno sve potrebne elemente za stvaranje zvuka upotrebom micro:bit uređaja, dovoljno je povezati navedene uređaje prema shemi prikazanoj na Slika 37. Povezane komponente prikazuje Slika 39.



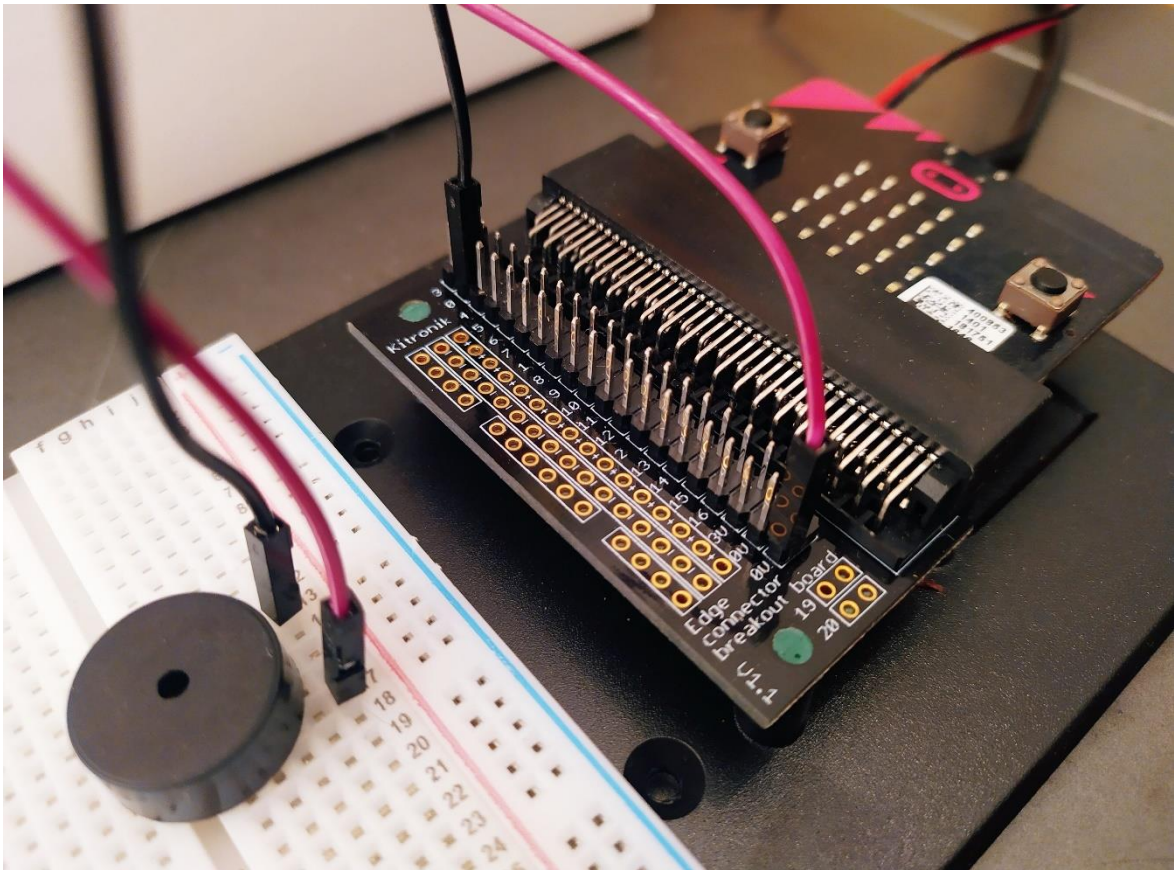
Slika 36 Piezoelektrični zvučnik



Slika 38 Breadboard



Slika 37 Shema povezivanja piezoelektričnog zvučnika



Slika 39 Micro:bit za dojavu alarma povezan sa zvučnikom, upotrebom dodatka za proširenja, sadržanih u Inventor's kitu

4 Zaključak

U radu smo dali pregled nekih od brojnih mogućnosti micro:bita, mikroračunala koje je prije 5 godina razvijeno za potrebe suvremenog obrazovanja i danas se široko koristi u europskom obrazovanju.

Suvremena nastava fizike u svom središnjem dijelu mora imati fizikalni pokus koji je izvrstan alat za motivaciju učenika i aktivno uključivanje kroz istraživanja u nastavni proces.

Micro:bit otvara mogućnost razvoja brojnih istraživačkih ideja u nastavi fizike uz relativno male troškove. Upotrebom micro:bit uređaja kao mjernog instrumenta u fizikalnom pokusu uz mogućnost prikupljanja i bežičnog slanja podataka, učenike uvodimo u istraživački proces. Premda je preciznost mjerenja manja nego s profesionalnom opremom, ona je dovoljna za detekciju i analizu fizikalnih pojava. Na nastavnicima fizike i drugim stručnjacima u obrazovanju je zadatak osmisliti brojne problemske ili istraživačke zadatke u nastavi fizike koji bi učenika kao istraživača stavili u središte nastavnog procesa. Micro:bit kao alat pri tome otvara brojne mogućnosti. Učinkovitost ovakvog pristupa u stjecanju trajnih fizikalnih znanja i vještina puno je veća nego da se koristi sofisticirana laboratorijska oprema koju učenici najčešće percipiraju kao crne kutije bez dubljeg razumijevanja fizike.

Kada govorimo o micro:bitu, odnosno mikroračunalima naglasit ćemo još jednom njihov značaj u izvođenju kvantitativnih mjerenja u fizikalnom pokusu i prikupljanja podataka (ponekad i kroz dulja razdoblja od više sati, dana ili tjedana). Izmjereni podaci se prenose na računalo te se, ovisno o razini obrazovanja na kojoj se tema obrađuje, mogu izvoditi jednostavne i složene analize, pa i fizikalna modeliranja. Pored micro:bita koji je dizajniran za rano učenje STEM vještina u osnovnoj školi, u srednjoj školi i na fakultetima se mogu koristiti i kompleksnije arhitekture poput Arduina i Raspberry pi koje otvaraju mogućnost i profesionalnih primjena (mjerni instrumenti i senzori bazirani na tim arhitekturama). Moglo bi se reći da je micro:bit i prvi korak u smjeru razvoja i dizajna novih uređaja ili visokotehnoloških proizvoda. U obrazovanju 21. stoljeća važno je da se učenici praktično upoznaju s tim procesom.

5 Literatura

- Burk, J. (n.d.). *Measuring the length of a pendulum with a microbit*. Dohvaćeno iz Quantum progress: <https://quantumprogress.wordpress.com/2019/06/12/measuring-the-length-of-a-pendulum-with-a-microbit/>
- Edutorij. (n.d.). *Matematičko njihalo*. Dohvaćeno iz Harmonijsko titranje: <https://edutorij.e-skole.hr/share/proxy/alfresco-noauth/edutorij/api/proxy-guest/a46bb23b-608e-45b5-b7f6-c952a83441fa/matematicko-njihalo.html>
- Landoni, B. (n.d.). *BBC micro:bit*. Dohvaćeno iz BBC micro:bit: <https://www.open-electronics.org/bbc-microbit/>
- Lyman, C. (Veljača 2018). *Data Collection Methods Microbit Science*. Dohvaćeno iz <https://makecode.microbit.org/courses/ucp-science/data-collection>
- makers, C. (n.d.). *BBC microbit STEM revolucija*. Dohvaćeno iz Croatian makers: <https://croatianmakers.hr/hr/stem-revolucija/>
- MONK, S. (2019). *Micro:bit for mad scientists : 30 clever coding and electronics projects for kids*. Dohvaćeno iz <https://lcn.loc.gov/2019015785>
- Multiwingspan. (n.d.). *BBC micro:bit*. Dohvaćeno iz BBC micro:bit: <http://www.multiwingspan.co.uk/micro.php?page=pycomp>
- Pintarić, G. (n.d.). *Upotreba mikrokontrolera u nastavi fizike*. Dohvaćeno iz Upotreba mikrokontrolera u nastavi fizike: http://www.hfd.hr/sns2017/wp-content/uploads/2017/04/Zbornik_radova_2017-kb-str241-245.pdf
- Runberg, D. (n.d.). *Free your radio functionality on the micro:bit*. Dohvaćeno iz Free your radio functionality on the micro:bit: <http://blog.sparkfuneducation.com/working-with-the-radio-on-the-microbit>
- Sengul, C., & Kirby, A. (2017). *knjiga*. Dohvaćeno iz Micro:bit nominet research: https://microbit.nometresearch.uk/networking-book-pdf/networking_with_the_microbit.pdf

6 Životopis

Josip Bakula, rođen 5. rujna 1991. god., u Zagrebu. Obrazovanje započeo u II. Osnovnoj školi Vrbovec, gdje se zahvaljujući iznimnim učiteljima i stvorio interes za fiziku, nastavio u srednjoj školi Tehnička škola Ruđera Boškovića, smjer tehničar za elektroniku, nakon koje je upisan studij fizike i informatike na Prirodoslovno-matematičkom fakultetu u Zagrebu. Privođenjem preddiplomskog studija kraju, spletom okolnosti stvorila se potreba za izdvajanjem većeg dijela raspoloživog vremena na praktičnu primjenu naučenoga u poslovima:

- Predavač fizike (pripreme za državnu maturu, 4.god. iskustva, POU Algebra)
- Predavač Digitalne akademije (u osnovnim školama, 3.god. iskustva, POU Algebra)
- Nastavnik informatike i matematike u osnovnoj školi (1,5 god)
- Pomoć u učenju fizike, matematike i elektrotehnike
- Izrada projekata sigurnosne zaštite u AutoCAD-u

2018. god. dodijeljeno priznanje za najboljeg nastavnika godine, u sklopu POU Algebra, prema rezultatima anketa ispunjavanja od strane polaznika priprema za državnu maturu.

Slobodno vrijeme ispunjava fotografijom, bicikliranjem i drugim oblicima aktivnog odmora.