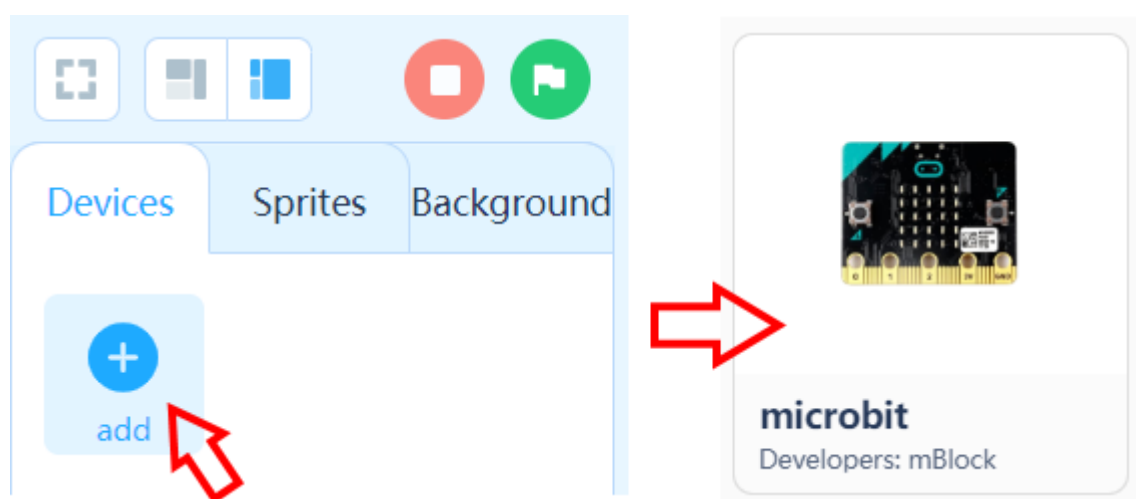


1. domaća zadaća

4 aktivnosti za Micro:bit

Priprema

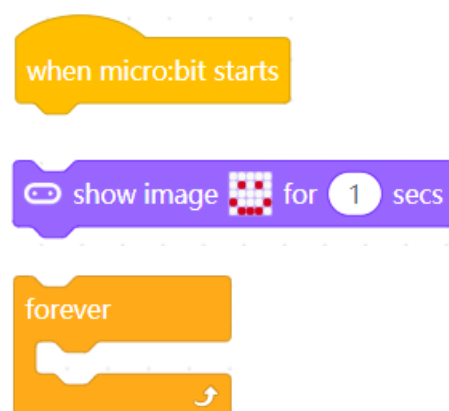
1. Spojimo *Micro:bit* s računalom pomoću microUSB kabela.
2. Pokrenemo *mBlock 5* (v5.1.0).
3. Dodamo novi uređaj i odaberemo *Micro:bit*.



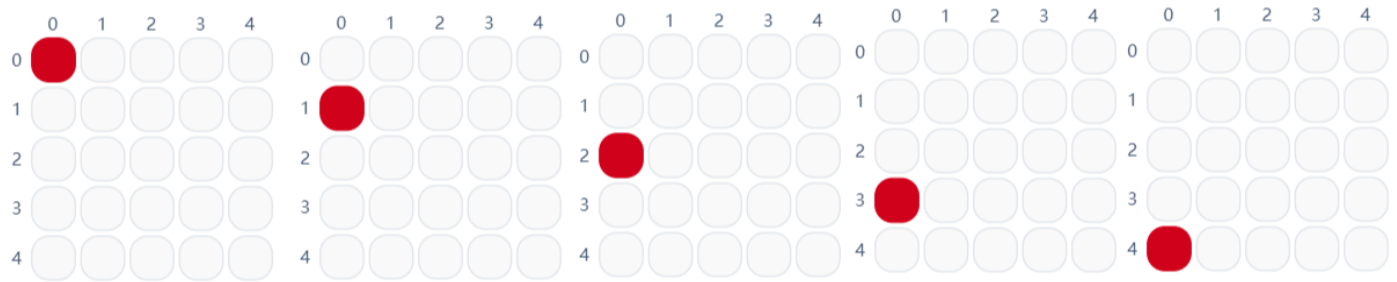
1. aktivnost „KIŠA“

U ovoj aktivnosti naučit ćemo kako napraviti animaciju na *Micro:bit*-ovoj mreži LED lampica. Izradit ćemo animaciju kiše.

Bit će nam potrebni sljedeći blokovi:



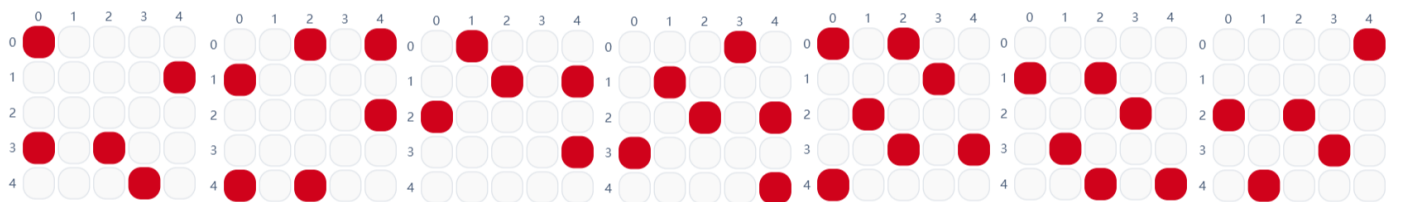
1. Pronađimo i umetnimo *početni blok* i *blok za prikaz slike*. Prvo ćemo nacrtati samo jednu kap kiše (1 lampica) te animirati njezin pad.



when micro:bit starts

```
show image [5x5 grid with 1 dot at (0,0)] for 0.15 secs
show image [5x5 grid with 1 dot at (1,0)] for 0.15 secs
show image [5x5 grid with 1 dot at (2,0)] for 0.15 secs
show image [5x5 grid with 1 dot at (3,0)] for 0.15 secs
show image [5x5 grid with 1 dot at (4,0)] for 0.15 secs
```

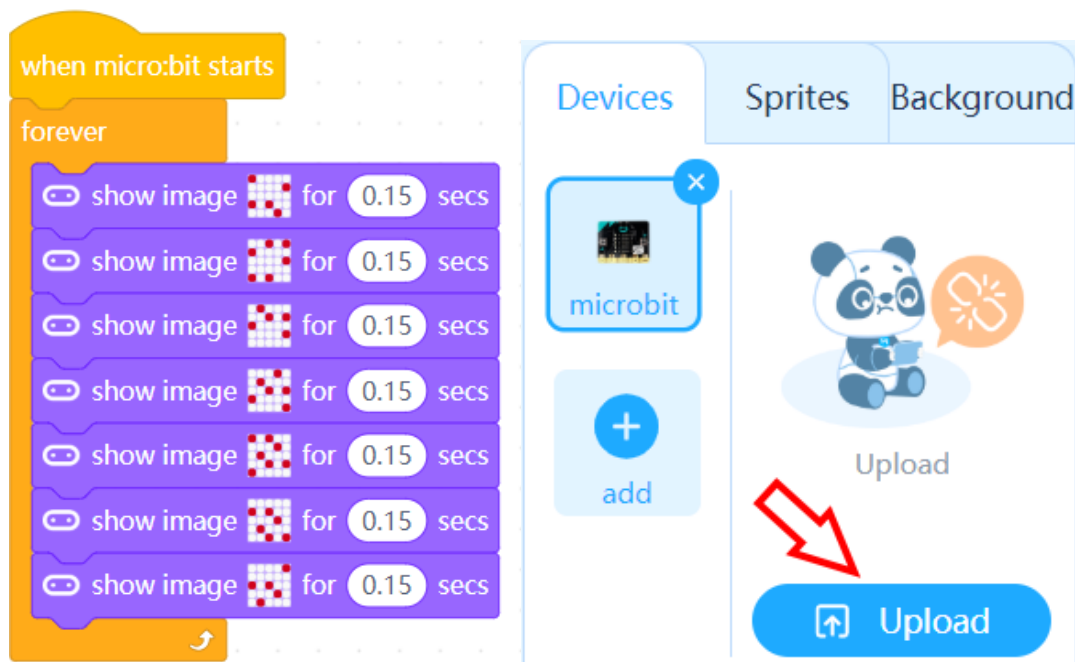
2. Na isti način dodat ćemo još nekoliko kapi, ali na različitim pozicijama kako bi se postigao efekt kiše. Odabirom vremena prikazivanja svake sličice postaviti ćemo brzinu padanja kiše.



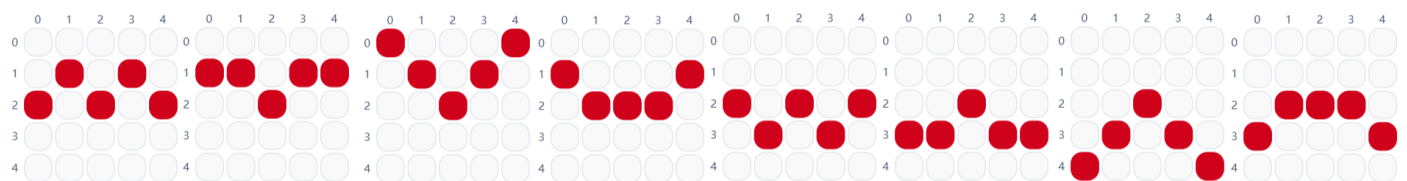
when micro:bit starts

```
show image [5x5 grid with 5 dots at (0,0), (1,1), (2,2), (3,3), (4,4)] for 0.15 secs
show image [5x5 grid with 5 dots at (0,0), (1,0), (2,1), (3,2), (4,3)] for 0.15 secs
show image [5x5 grid with 5 dots at (0,0), (1,1), (2,2), (3,3), (4,4)] for 0.15 secs
show image [5x5 grid with 5 dots at (0,0), (1,1), (2,2), (3,3), (4,4)] for 0.15 secs
show image [5x5 grid with 5 dots at (0,0), (1,1), (2,2), (3,3), (4,4)] for 0.15 secs
show image [5x5 grid with 5 dots at (0,0), (1,1), (2,2), (3,3), (4,4)] for 0.15 secs
show image [5x5 grid with 5 dots at (0,0), (1,1), (2,2), (3,3), (4,4)] for 0.15 secs
```

3. Kako kiša ne bi prestala padati dodat ćemo beskonačnu petlju. Da bi animaciju vidjeli na *Micro:bit*-u , trebamo napraviti *upload* programa.



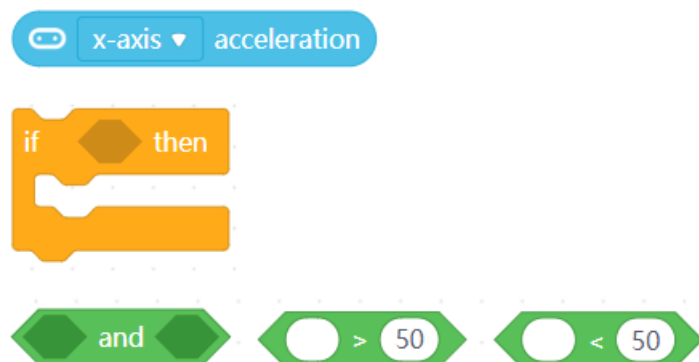
BOUNS ANIMACIJA



2. aktivnost „KIŠOBRAN“

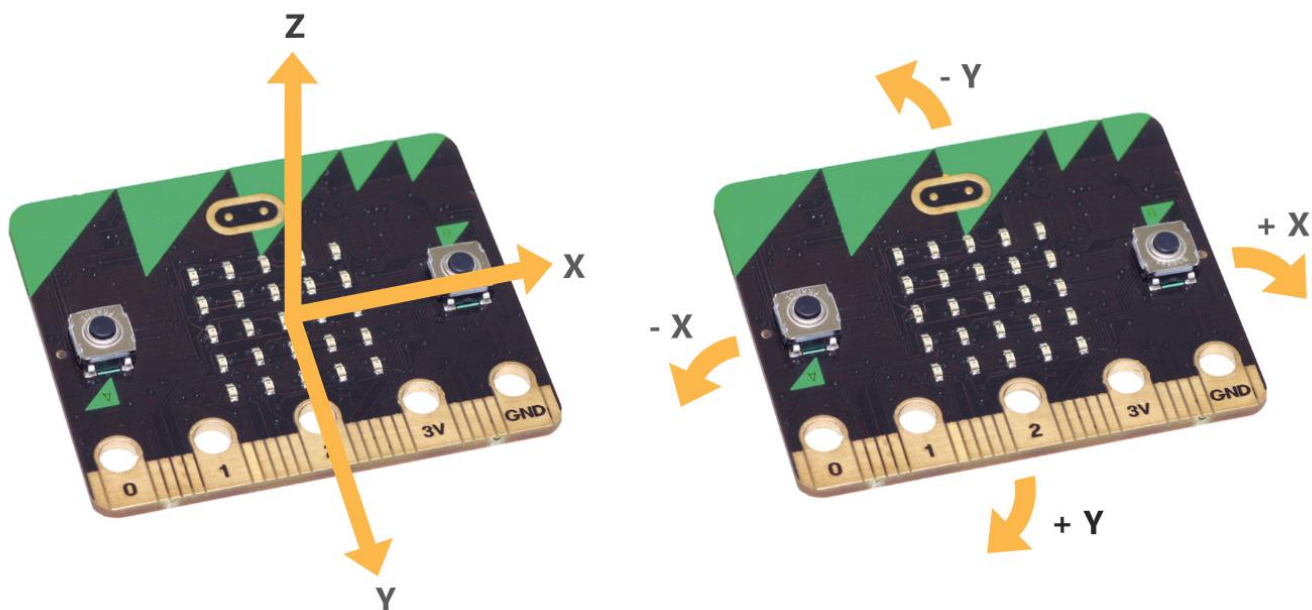
U ovoj aktivnosti ćemo naučiti nešto o žiroskopu koji je ugrađen u *Micro:bit* te kako pomoću njega kontrolirati animaciju.

Novi blokovi koji će nam biti potrebni:



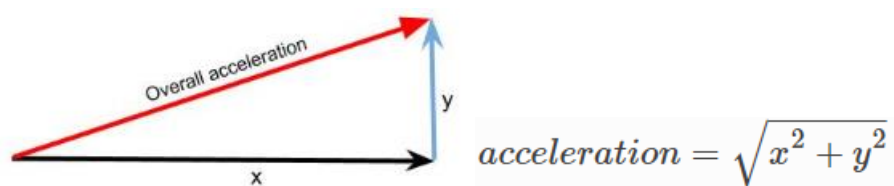
Žiroskop je uređaj koji pomoću Zemljine sile teže određuje orijentaciju. Pomoću njega možemo odrediti:

- Naginje li se *Micro:bit* lijevo ili desno – X-os
- Naginje li se *Micro:bit* naprijed ili nazad – Y-os
- Pomiče li se *Micro:bit* gore ili dolje – Z-os



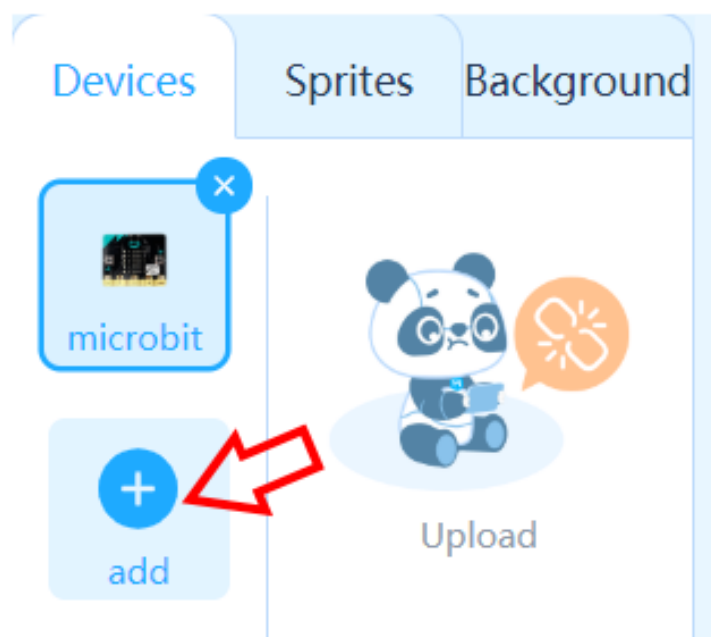
Žiroskop će nam dati vrijednost od -1024 do +1024. Npr. za x-os -1024 znači da je *Micro:bit* nagnut skroz lijevo, 0 znači da stoji ravno, a 1024 znači da je nagnut skroz desno, analogno i za ostale osi.

S druge strane, akcelerometar je uređaj koji mjeri ne-gravitacijsko ubrzanje. *Micro:bit* ne daje nikakve informacije o ubrzanju. One se mogu odrediti na osnovi podataka koje daje žiroskop.

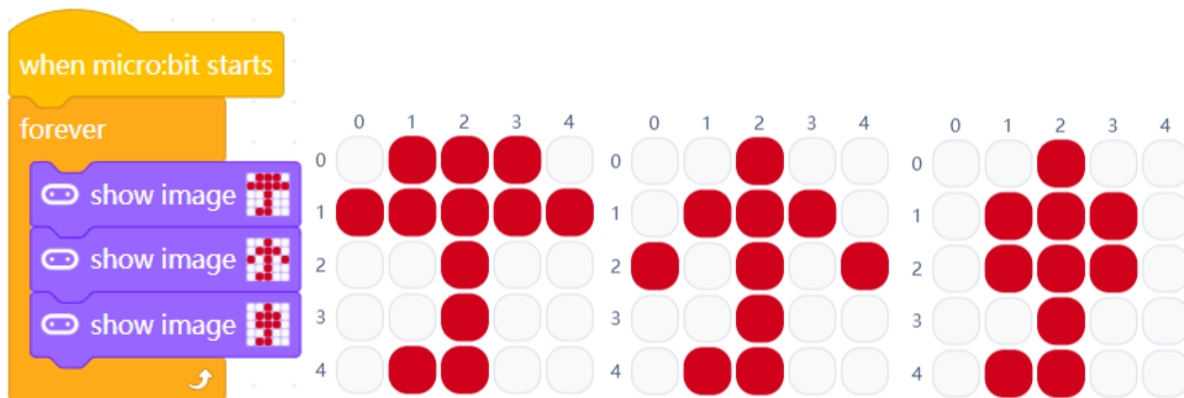


Dakle, naziv akcelerometar u nazivu bloka nije baš točan.

1. Dodat ćemo još jedan *Micro:bit* uređaj

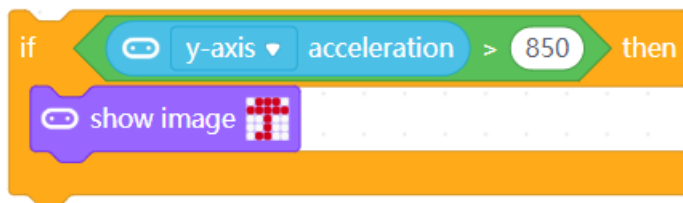


2. Iskoristimo ono što smo naučili u prethodnoj aktivnosti o animacijama i napravimo jednostavnu animaciju otvaranja i zatvaranja kišobrana. Budući da ćemo animaciju kontrolirati žiroskopom, tako da kišobran bude otvoren kada *Micro:bit* stoji uspravno, zatvoren kada *Micro:bit* stoji ravno, i polu otvoren između, odabrat ćemo blok za prikaz slike bez vremenskog ograničenja.

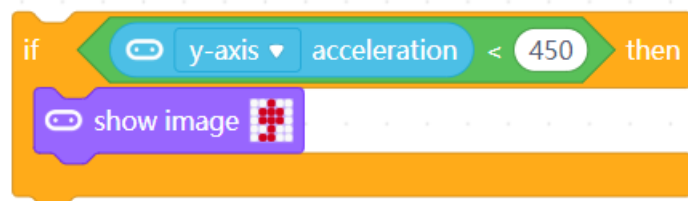


3. Sada ćemo za prikaz svake slike dodati uvjete kada želimo da se ona prikaže. Nas zanima kada se *Micro:bit* naginje prema naprijed, odnosno nazad. Za to treba promatrati očitavanje žiroskopa za y-os.

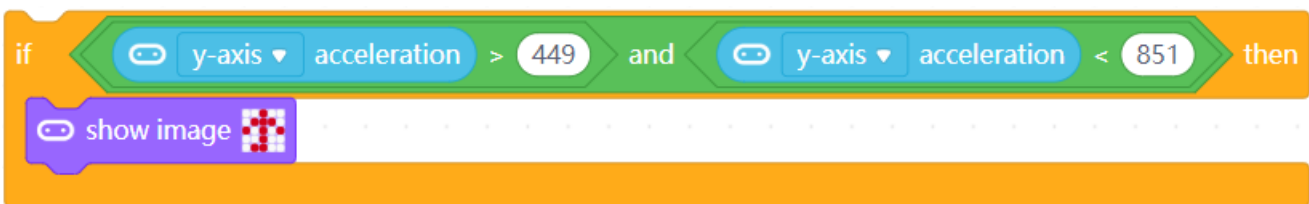
Kišobran je otvoren kada *Micro:bit* stoji uspravno, to je kada žiroskop za y-os daje vrijednosti blizu 1024. Mi ćemo dati malo više prostora, te ćemo gledati sve vrijednosti veće od 850.

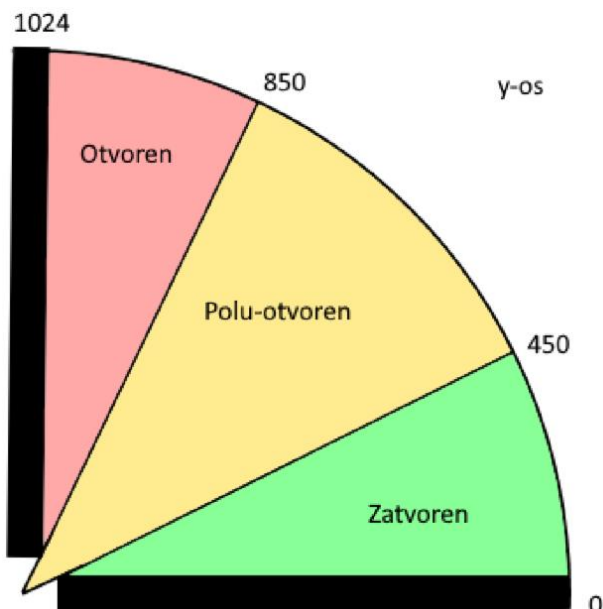


Slično kišobran je zatvoren kada žiroskop daje vrijednosti blizu 0. Mi ćemo gledati vrijednosti manje od 450.

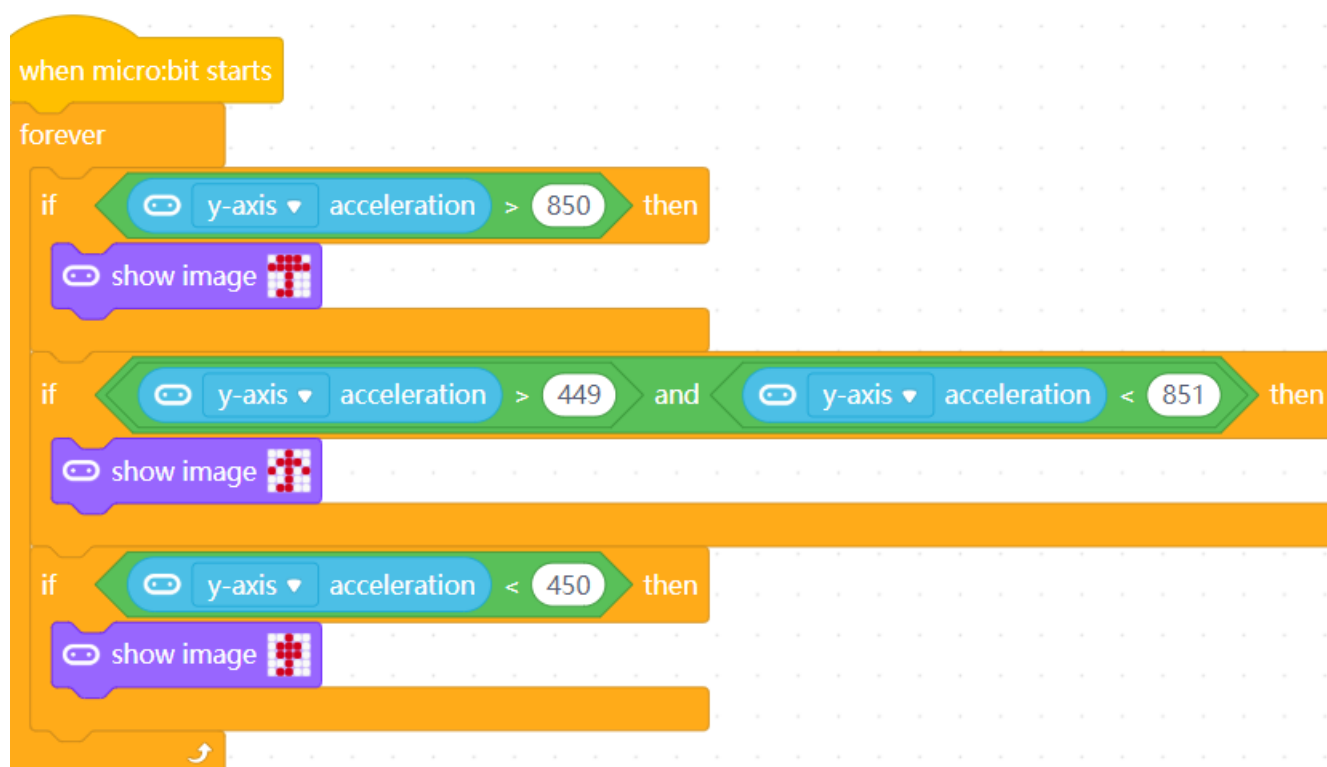


Za sve ostale vrijednost, veće od 449 i manje od 851 kišobran je poluotvoren.





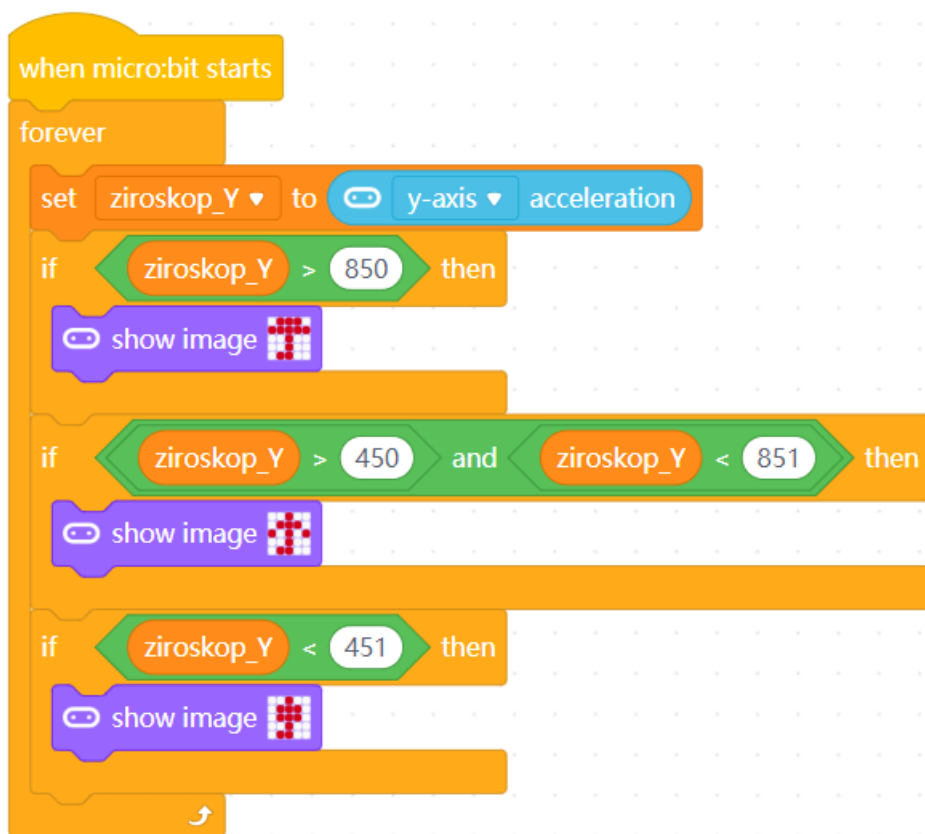
4. Posložimo sve blokove u konačni program spreman za *upload* na *Micro:bit*.



5. Primijetimo kako u našem programu čak 4 puta očitavamo vrijednost koju nam daje žiroskop. Umjesto toga mi možemo tu vrijednost učitati samo jedanput te ju zapamtiti u varijabli. Kreirajmo varijablu koja će pamtit i očitavanje sa žiroskopa i dajmo joj prikladno ime.



Na početku beskonačne petlje zapamtimo očitavanje žiroskopa pomoću varijable te u nastavku programa koristimo varijablu kada nam je potrebno očitavanje žiroskopa.

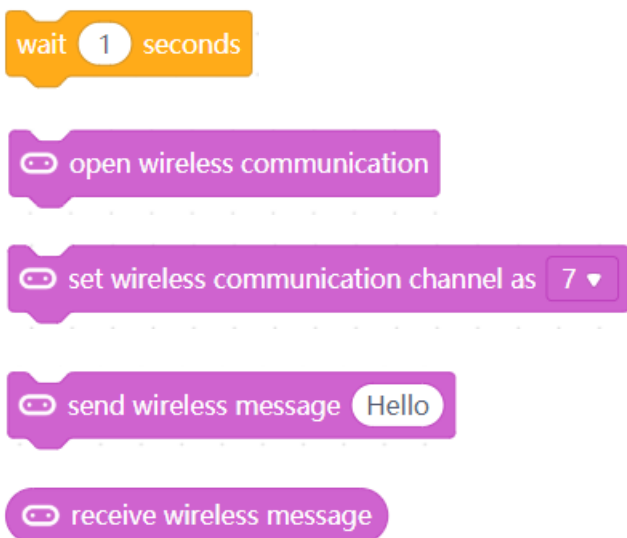


3. aktivnost „ISPOD MOG KIŠOBRANA“

U prethodnim aktivnostima napravili smo kišu i kišobran. U ovoj aktivnosti ćemo naučiti kako *Micro:bit*ovi mogu međusobno komunicirati kako bi kišobran zaustavio kišu.

Za ovu aktivnost bit će potrebna dva *Micro:bit*a, u daljnjem tekstu, *Kišobit* i *Branbit*.

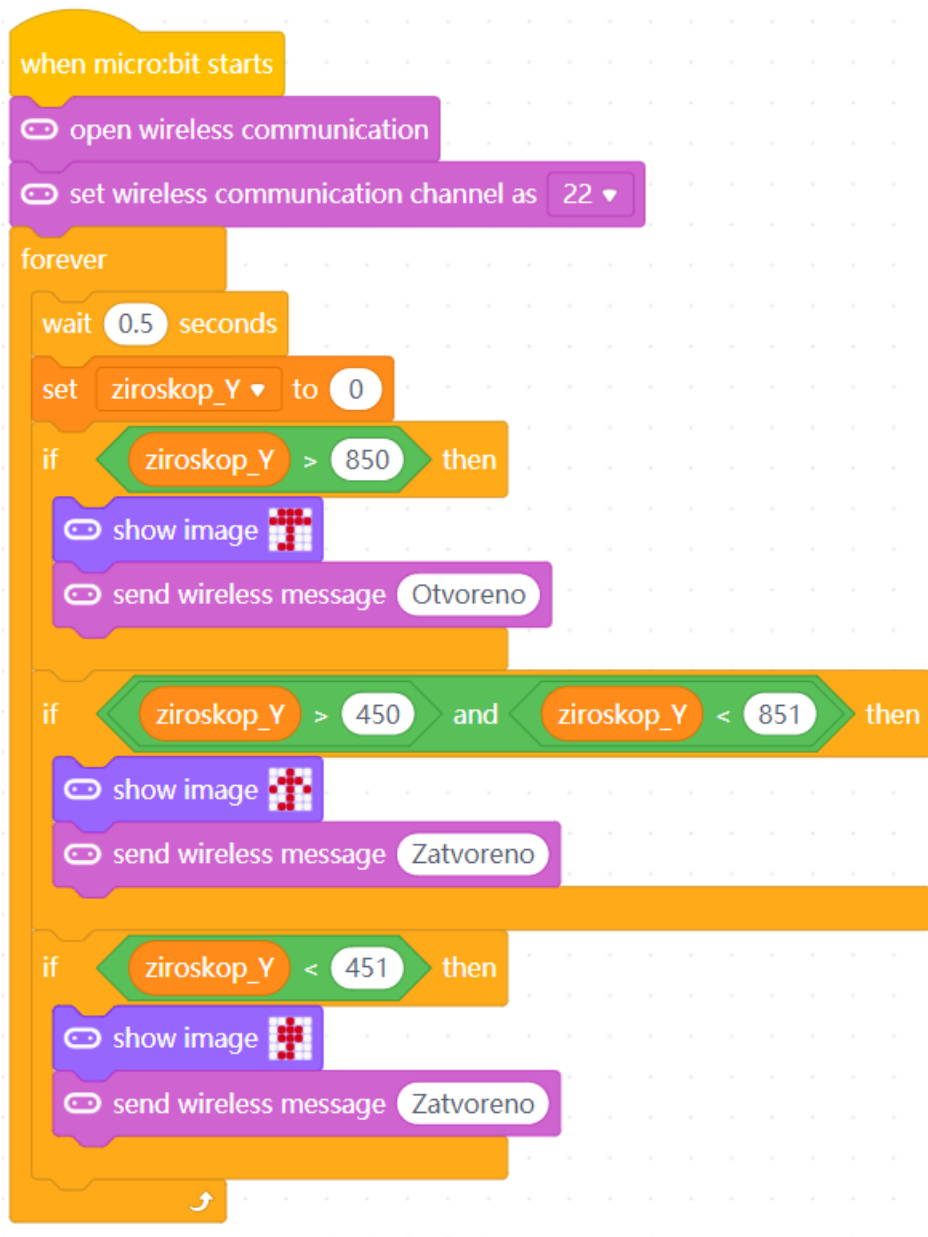
Novi blokovi u ovoj aktivnosti:



1. Prvo ćemo preraditi kod iz 2. aktivnosti koji ćemo postaviti na *Branbit*. Kako bi *Branbit* uspješno komunicirao s *Kišobitom* potrebno je prvo otvoriti bežičnu komunikaciju te odabrati jedan od kanala. Važno je da i *Kišobit* i *Branbit* imaju odabrani isti kanal. Mi ćemo odabrati kanal 22.

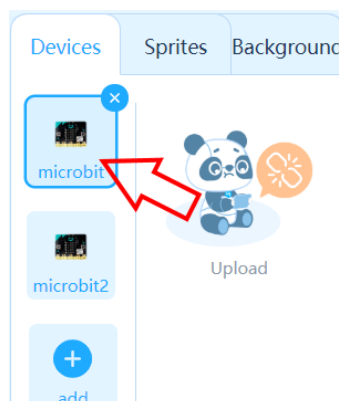
Branbit treba javiti Kišobitu kada je otvoren i kada je zatvoren kako bi Kišobit znao kada treba zaustaviti, a kada pustiti kišu. Pomoću bloka za slanje poruke poslat ćemo odgovarajuću poruku za svaku poziciju kišobrana.

Dodat ćemo i blok koji će pričekati pola sekunde unutar beskonačne petlje kako ne bi prebrzo slali previše poruka.



2. Kod ćemo *uploadati* na Branbit, ištekat ga, i s računalom spojiti Kišobit.

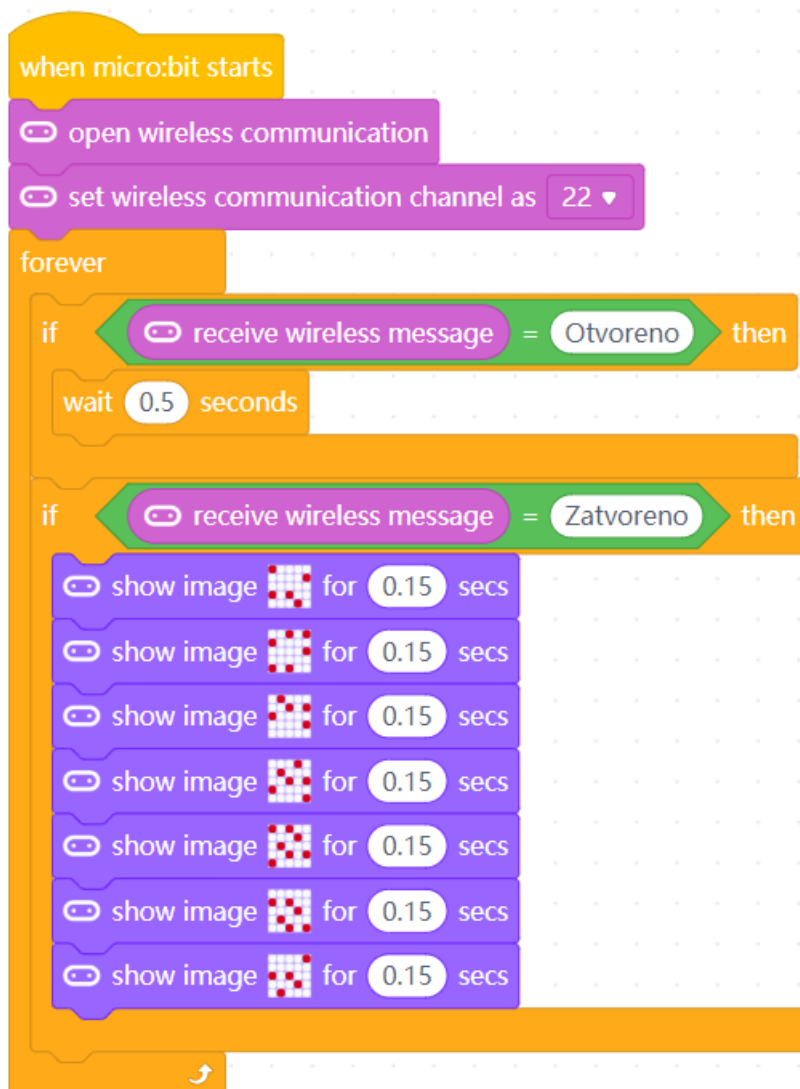
3. Vratimo se na program iz 1. aktivnosti.



Branbit će poslati poruku na kanalu 22, na kojem je Kišobit treba primiti. Opet ćemo otvoriti bežičnu komunikaciju te nakon toga odabrati kanal 22. Ovaj put ćemo primiti poruke te ovisno o njihovom sadržaju upravljati kišom.

Ako primimo poruku „Otvoreno“ znači da je kišobran otvoren, u tom slučaju ćemo samo pričekati pola sekunde tj. nećemo pustiti animaciju kiše.

Ako primimo poruku „Zatvoreno“, kišobran je zatvoren, u tom slučaju kiša može padati, te trebamo pustiti animaciju kiše.

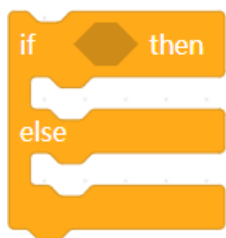


4. Napravimo *upload* programa na Kišobit. Uštekamo i Branbit kako bi se i on uključio. Testiramo program.

4. aktivnost „KIŠNA GLISTA“

U ovoj aktivnosti ćemo iskoristiti ono što smo do sada naučili o animacijama i žiroskopu. Naučit ćemo kako dodavati varijable i izraditi vlastite blokove.

Novi blokovi koji će nam biti potrebni:



Make a Variable

Make a Block

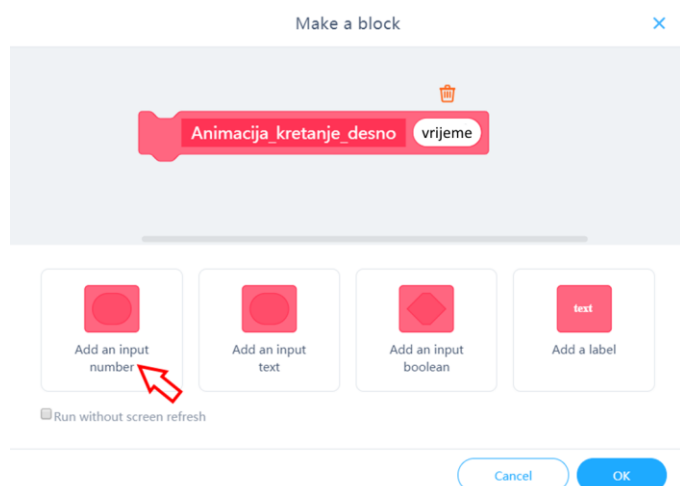
1. U ovoj aktivnosti želimo izraditi kišnu glistu koja će se kretati prema lijevo kada *Micro:bit* nagnemo ulijevo, prema desno kada ga nagnemo udesno, i mirovati kada ga *Micro:bit* stoji ravno. Prvo ćemo izraditi animacije za svaku od mogućih stanja kišne gliste.

Budući da za animacije koristimo puno blokova i to zauzima puno mjesta u kodu našeg programa mi možemo izraditi svoj novi blok koji će biti zadužen za određenu animaciju te pomoću njega dodati animaciju u program kao jedan blok. Na taj način animaciju možemo vrlo lako postaviti i na više mjesta unutar našeg programa.

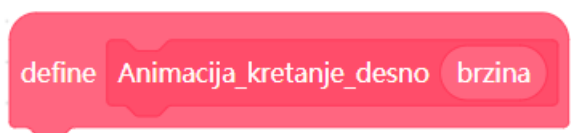
Kako bi izradili svoj blok potrebno je kliknuti na:



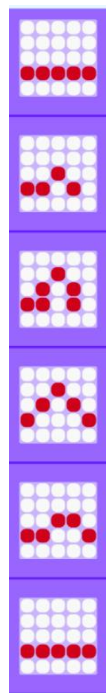
Nakon dobivamo sučelje za izradu bloka u kojem možemo dodavati razne „kućice“ koje želimo da naš blok ima. U prvu tekstualnu „kućicu“ upisat ćemo što naš blok radi te ćemo dodati još jednu brojčanu „kućicu“ u kojoj ćemo našem bloku zadati vrijeme trajanja svake sličice i nazvati je *vrijeme*.



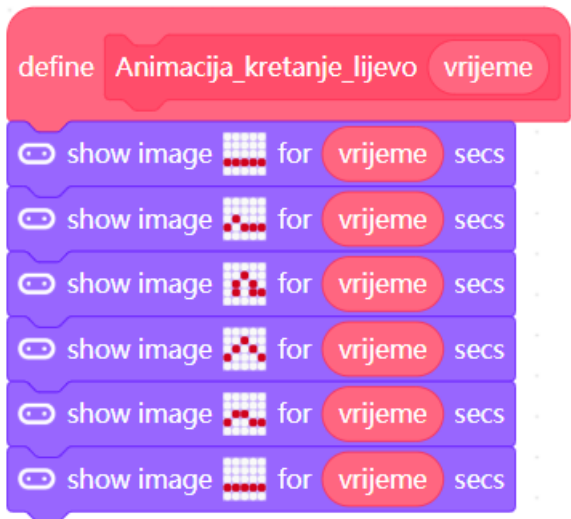
Nakon što kreiramo blok pojaviti će se sljedeći blok te će automatski biti generirana varijabla *brzina* koju možemo koristiti samo unutar našeg bloka.



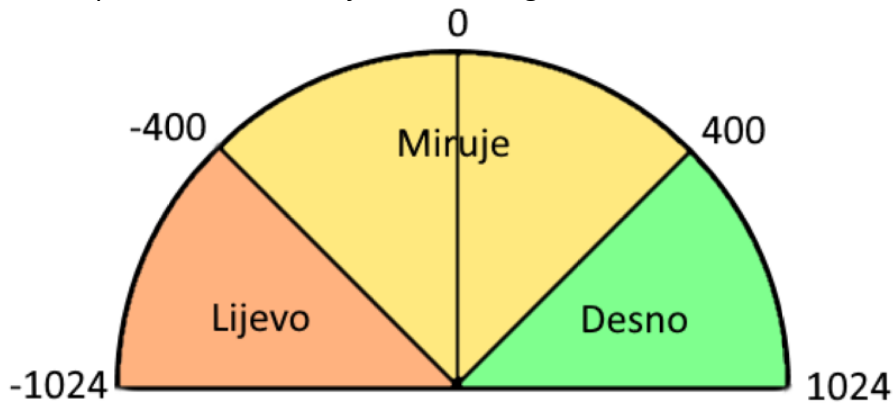
Na taj blok možemo dodati blokove koje želimo da on sadržava. U našem slučaju animaciju kretanja gliste prema desno u kojoj se svaka slička prikazuje onoliko sekundi koliko je pohranjeno u varijabli vrijeme.



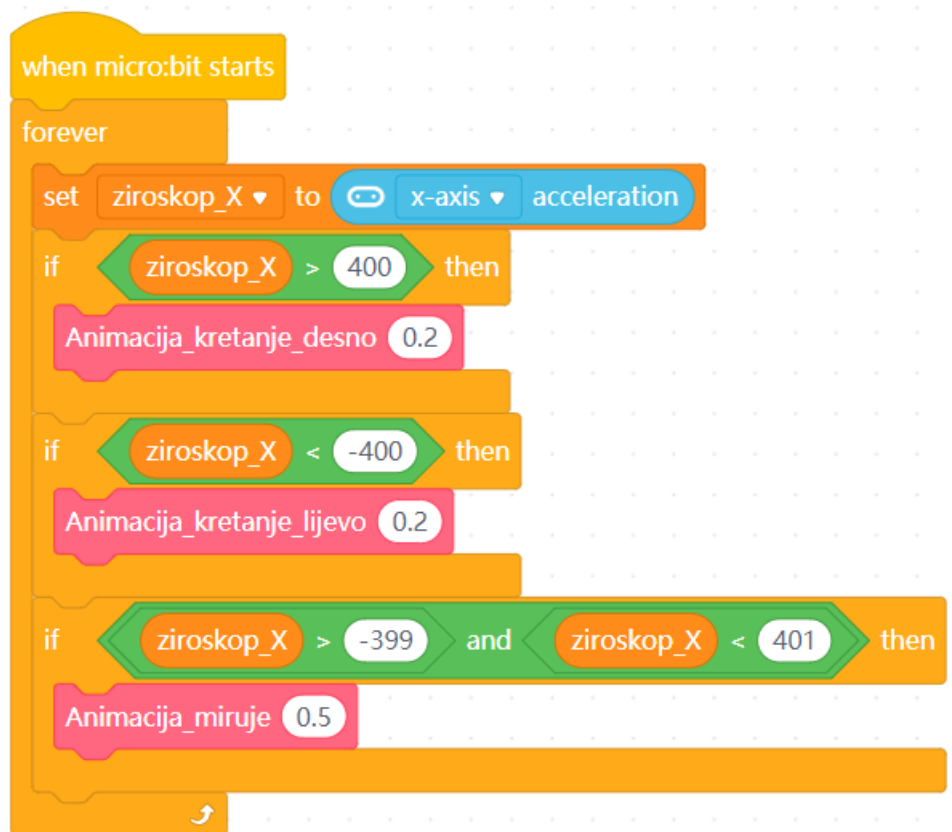
Napravimo i ostale animacije.



2. Dodajmo sada upravljanje glistom. Slično kao u prethodnoj aktivnosti, ali sada nam je potrebna os x žiroskopa kako bi utvrdili lijevi i desni nagib *Micro:bit*-a.



Umjesto konkretnih animacija sada koristimo nove blokove koje smo kreirali. Te u njima zadajemo brzinu izvođenja animacije.



3. Poznata je činjenica da se kišna glista kreće brže kada pada kiša. Kada otvorimo kišobran kiša ne pada. Kako bi glista znala kada mora usporiti i ona mora slušati poruke na kanalu 22. Dodajmo bežičnu komunikaciju i kontrolu brzine gliste. Za to ćemo trebati dodati još jednu varijablu pomoću koje ćemo postaviti brzinu gliste prema potrebi.

