

Spencer coding – first steps

ievads

Uzstādīšana

Paldies ka atbalsti CircuitMess un esi sveicināts Spencer programmēšanas apmācībā.

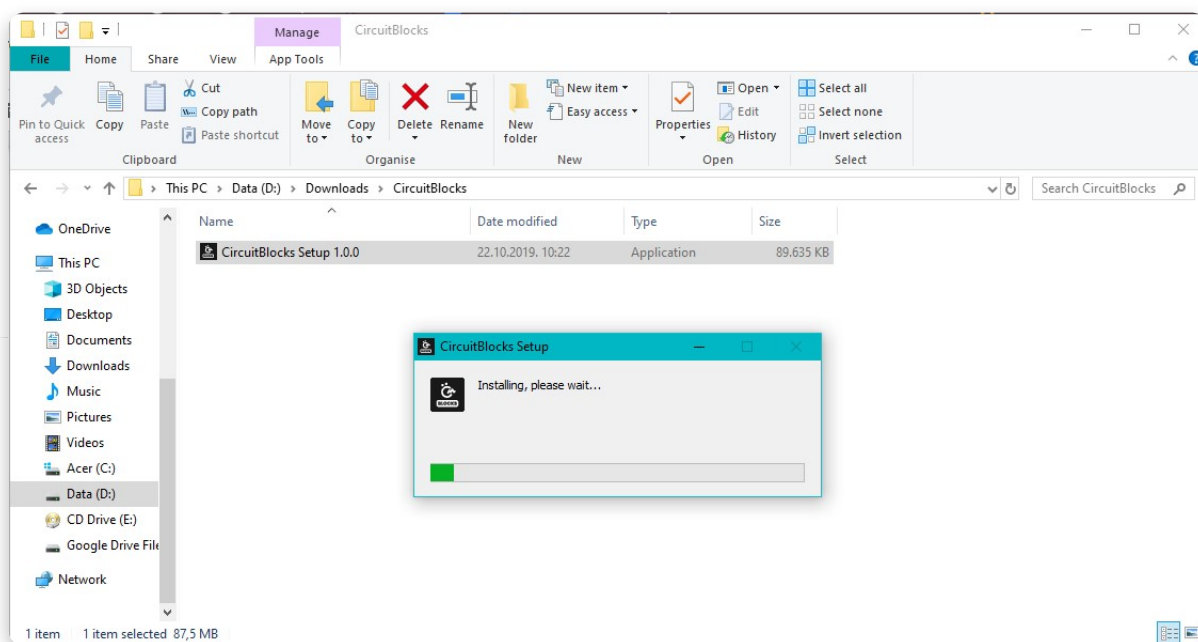
Mēs izmantosim CircuitBlocks lai programmētu tikko salikto balss asistentu. CircuitBlocks ir mūsu radīta programmēšanas aplikācija ko esam izstrādājuši. Spencer būs jāprogrammē grafiskajā blokshēmu programmēšanas vidē, kas jums palīdzēs iemācīties pirmos datorprogrammēšanas soļus.

Uzstādīšana

CircuitBlocks pašlaik darbojas un tādām platformām kā- Windows, Linux un Mac OS.

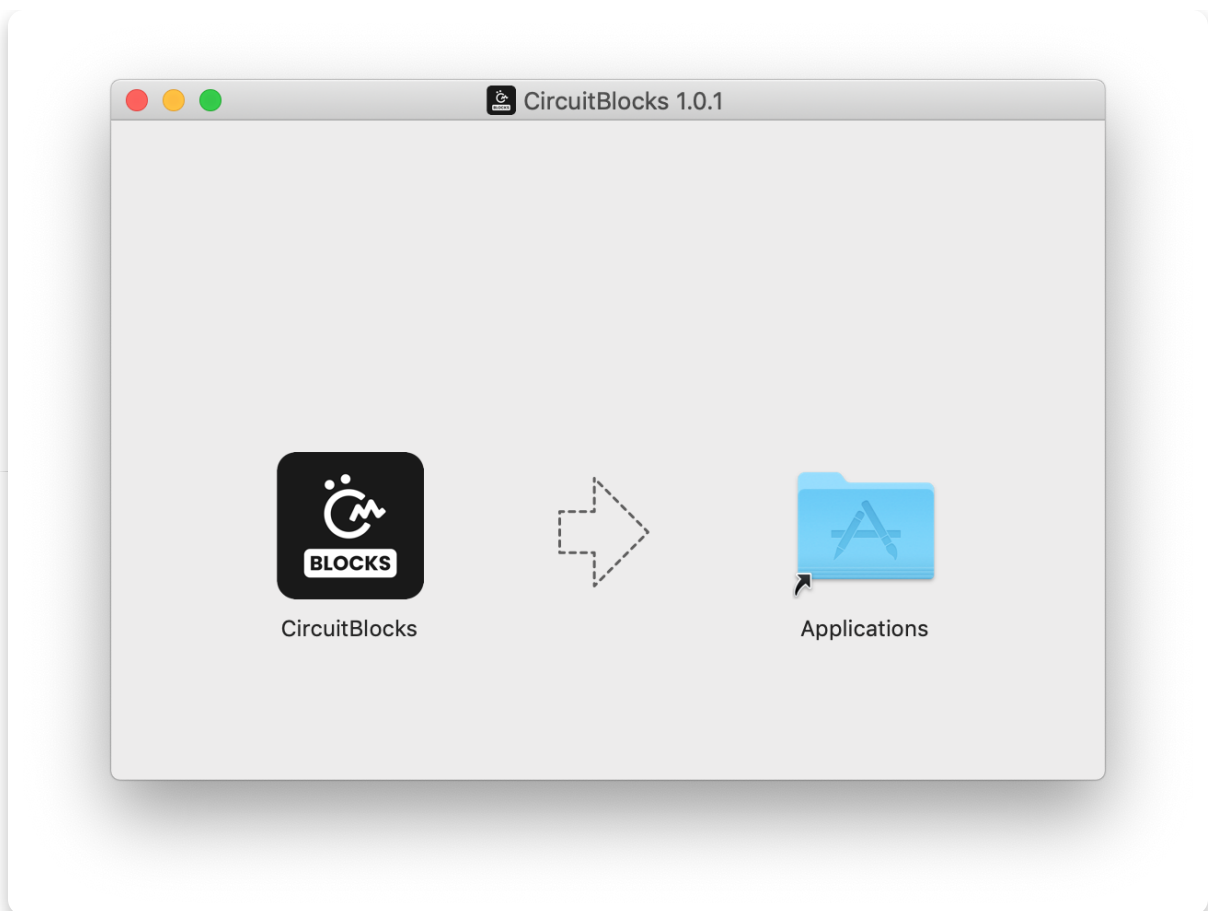
Ja jums ir Windows dators

1. Dodies uz <https://circuitmess.com/circuitblocks/download/>
2. Lejupielādē jaunāko versiju priekš Windows
3. Nospied uz faila "CircuitBlocks"]
4. CircuitBlocks automātiski tiks instalēts



Ja jums ir Mac dators

1. Dodies uz <https://www.circuitmess.com/circuitblocks/download>
2. Lejupielādē jaunāko CircuitBlocks priekš Mac OS
3. Pārviesto failu uz "Applications" mapi
4. CircuitBlocks tiks instalēts automātiski.



Ja jums ir Linux dators

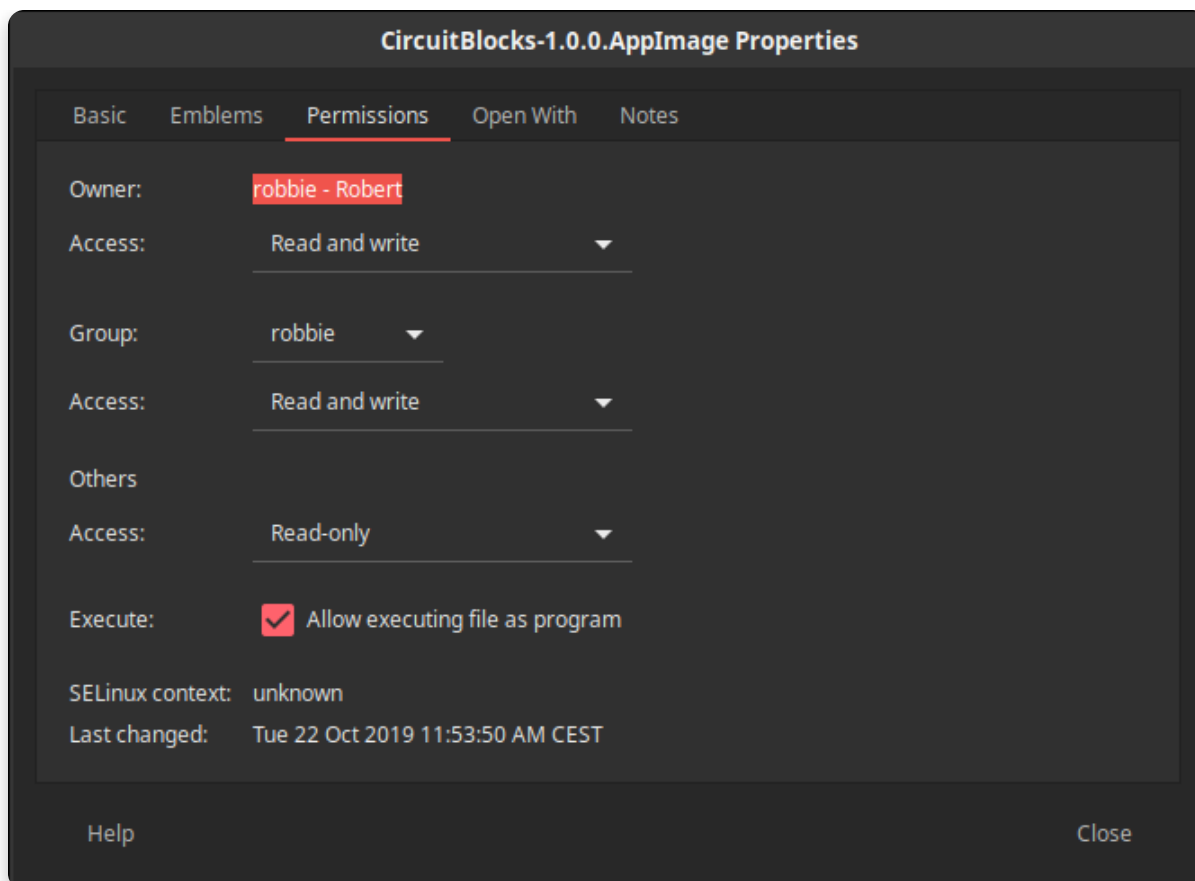
Ir 2 veidi kā instalēt CircuitBlocks uz Linux

“Vienkāršā” instalācija:

1. **Dodies uz** <https://www.circuitmess.com/circuitblocks/download>
2. **Nospied “Linux 64-bit” lejupielādes pogu**
3. Dubultklikšķis uz faila lai sāktu instalāciju (Ubuntu)
Vai
Atver termināli un ieraksti `-i <path to the downloaded file .deb>` (citām linux versijām)

AppImage instalācija:

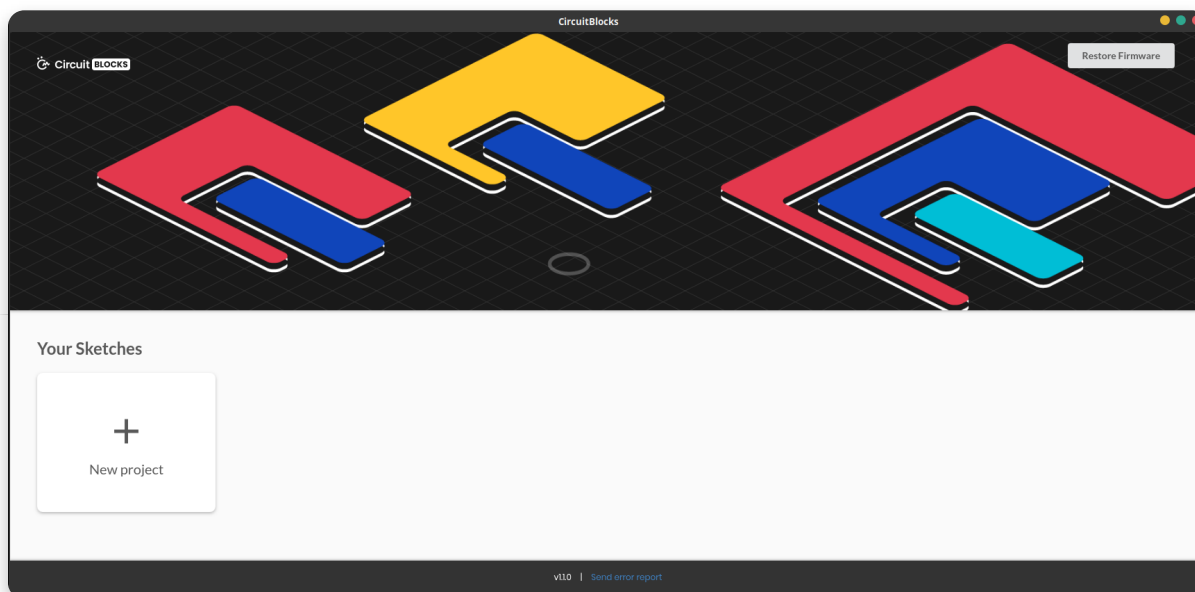
1. **Dodies uz** <https://www.circuitmess.com/circuitblocks/download>
2. **Nospied “Linux AppImage” lejupielādes pogu**
3. Nospied uz faila labo taustiņu un izvēlies “Properties”
4. Dodies uz “**Permissions**” un atzīmē “**Allow executing file as program**”
5. Dubultklikšķis uz instalācijas un tā sāksies automātiski



Ja saskaraties ar kādu problēmu kas saistīta ar instalāciju, lūdzu sazinaties ar mums **contact@circuitmess.com** un atsūtiet ekrānšāviņu un informāciju par savu problēmu.

ūsPamati

Lietotāja interfeiss



Kad atvēriesiet CircuitBlocks, redzēsiet logu kas izskatās šādi.

Tas ir pavisam vienkāršs – Lai sāktu jaunu projektu nospiediet “New project” pogu.

Saglabātie projekti tiks glabāti tur pat blakus “New project” pogai.

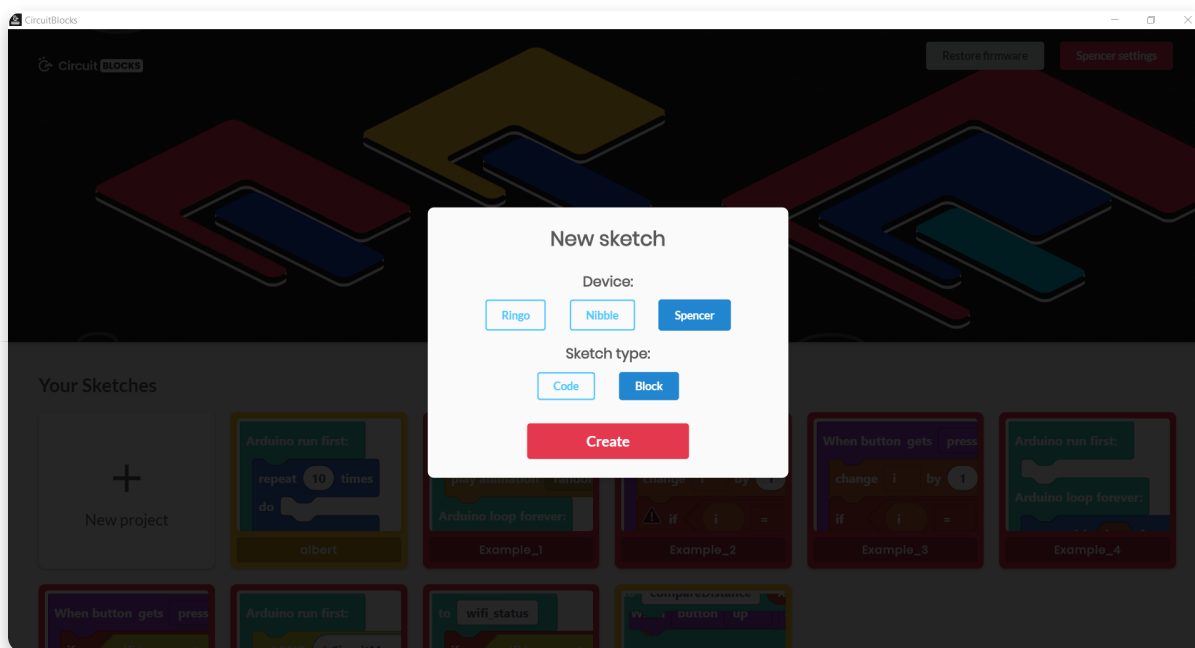
Ja jūs saskaraties ar kādu problēmu lietojot CircuitBlocks, nospiediet “Send error report” loga apakšā. Jums tiks parādīts kļūdas numurs. Lūdzu sazinaties ar mums un norādiet kļūdas numuru.

Jauna projekta izveide (sketch)

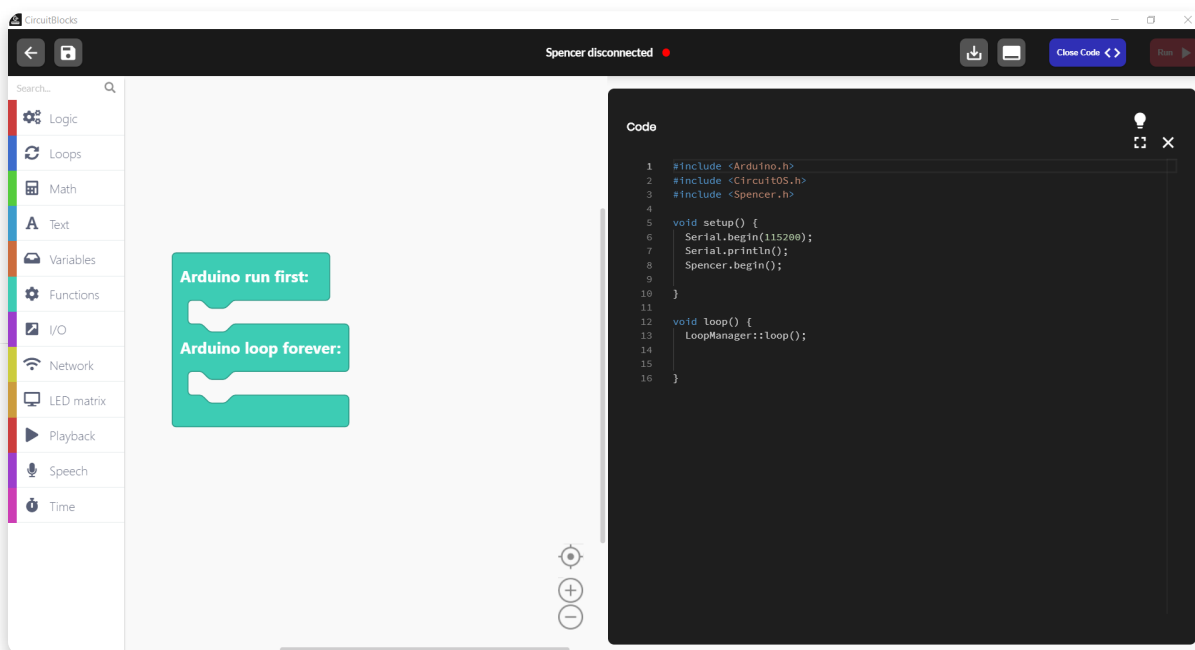
Nospied pogu "New project"
Jums jāizvēlas ierīce un programmēšanas veids.
Pie Device nospiediet **Spencer**.

Pie Sketch nospiediet **Block**.

Tālāk nospied Create pogu.



Tad atvērsies attēlā redzamais logs



Loga augšējā daļā atradīsiet rīkjoslau ar dažādām pogām.

Bloku izvēlnes rīks atrodas pašā kreisajā pusē- jūs varat izvēlēties blokus un pavilkt tos programmas izveides logā.

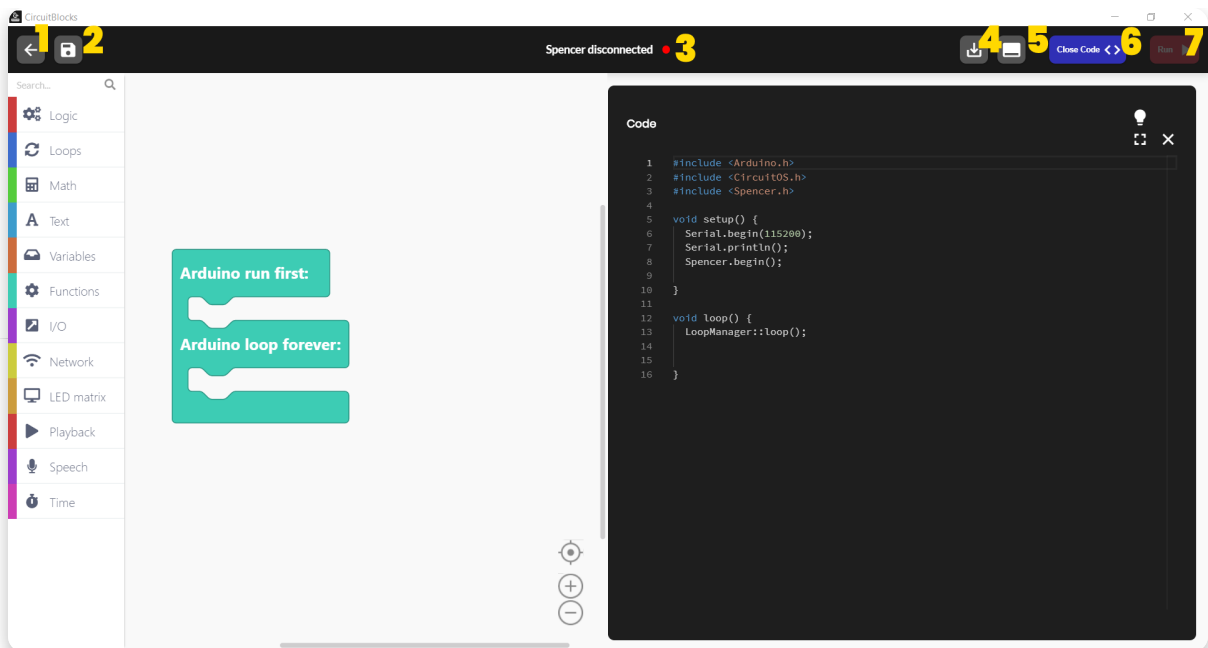
Šajā logā jūs veidosiet programmu no krāsainajiem blokiem.

Labajā ekrāna daļā jūs redzēsiet kodu kas rakstīts C++ tiklīdz veidosiet kodu ar blokiem.

C++ ir viena no populārākajām programmēšanas valodām, bet tā nav vienkārša ja iepriekš neesi saskāries ar to.

Tāpēc mēs esam radījuši CircuitBlocks – tu vari veidot kodu no krāsainajiem blokiem un paralēli redzēt kodu C++. Ka pietiekami daudz zināšanas, tad varēsi kodu veidot uzreiz C++ vidē.

Rīkjoslau



Šī ir īsa pamācība par to ko nozīmē pogas rīkjoslā.

1. **Atpakaļ uz galveno izvēlni** – atgriezies uz sākuma logu bez projekta saglabāšanas.
2. **Saglabāt/Saglabāt kā** – Saglabā jūsu projektu. Pārliecinies ka ik pa laikam saglabā savu darbu, kā arī pirms CircuitBlocks aizvēršanas.
3. **Spencer pieslēguma indikators** – Sarkanais punkts kļūst zaļš kad tavš Spencer ir pieslēgts pie datora ar USB kabeli.
4. **Bin faila eksportēšana** – Šī funkcija nodrošina Bin faila (programmas koda) saglabāšanu jūsu datorā.
5. **Serial monitor** – šī poga atver logu, kurā redzēsiet visus komunikācijas datus kas notiek starp Spencer un jūsu datoru.
6. **Close Code** – Ar šo pogu, jūs varat aizvērt vai atvērt logu kurā redzams C++ kods. Jūs to varat aizvērt, lai iegūtu vairāk vietas blokiem.
7. **Run** – Šī poga pārveidos kodu ko izveidojāt, par Spencer balss asistentam saprotamu kodu un nosūtīs to uz pašu Spencer.

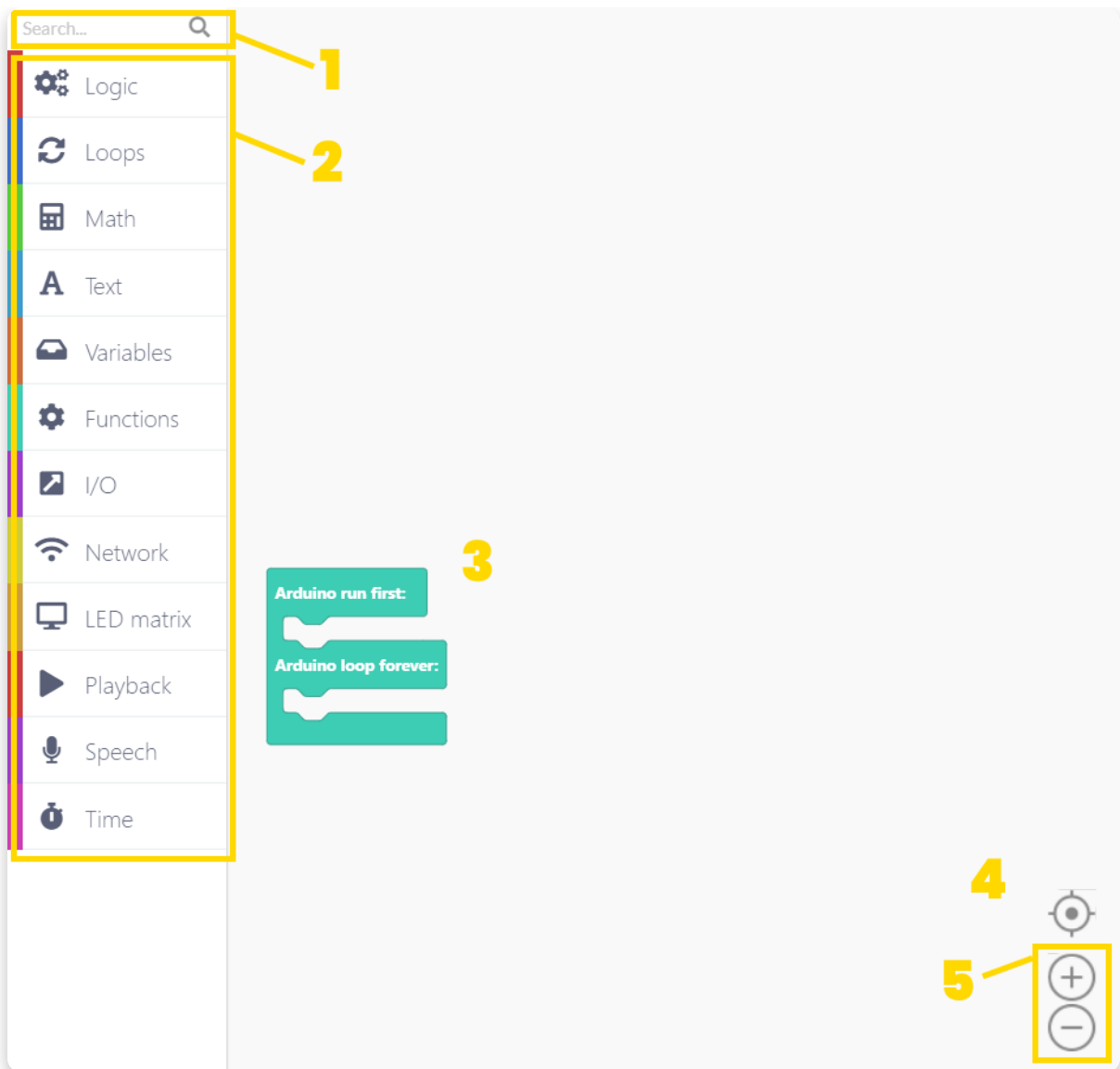
Koda logs

```
Code
1  #include <Arduino.h>
2  #include <CircuitOS.h>
3  #include <Spencer.h>
4
5  void setup() {
6      Serial.begin(115200);
7      Serial.println();
8      Spencer.begin();
9      Spencer.loadSettings();
10
11 }
12
13 void loop() {
14     LoopManager::loop();
15
16 }
17 }
```

Tā sauktais “Koda logs” sastāv no sekojošajām daļām;

1. **Primārais koda logs** – kods kas uzrakstīts ar C++ parādīsies šajā zonā kad sāksiet likt kopā blokus. Pamanīsiet ka dažādas koda daļas iekrāsosies dažādās krāsās. Tas ir paredzēts lai programmētājs vieglāk varētu uztvert dažādas koda daļas.
2. **Gaišās/Tumšais režīms** – jūs varat pārslēgt pamata un teksta krāsas un baltu vai melnu.
3. **Expand** – palielina kodu pa visu ekrānu. Nospied velreiz lai atkal to samazinātu.
4. **Aizvērt** – Aizver koda logu. Tāda pati funkcija kā “Close code” pogai rīkjoslā.

Bloku laukums



Šajā laukumā notiek koda veidošana.

Tas sastāv no sekojošajām daļām:

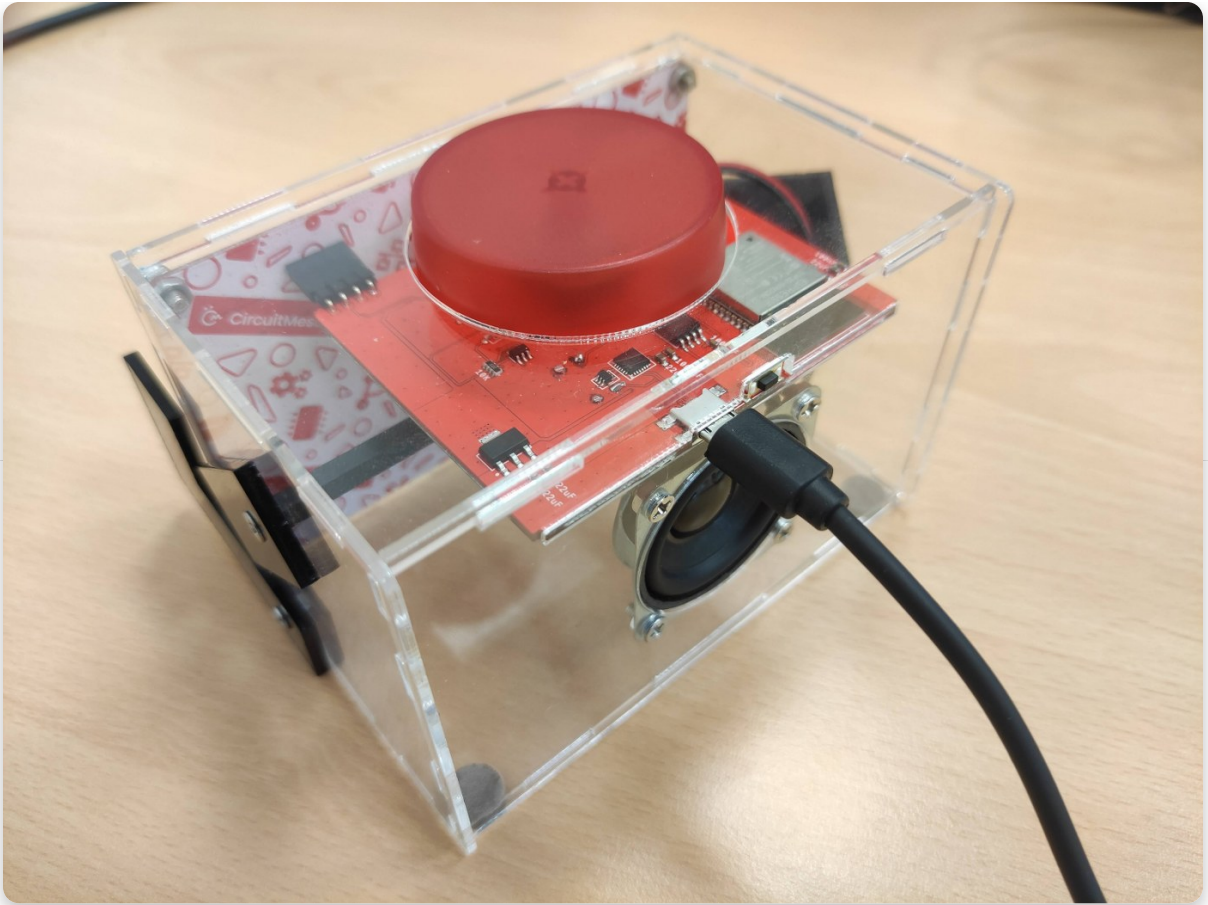
1. **Meklēšanas rīks** – meklē blokus pēc nosaukuma
2. **Bloku veidi** – Šeit bloki ir sadalīti pēc kategorijas. Katra kategorija ir nodalīta savā krāsā.
3. **Veidošanas zona** – Šajā zonā jūs varat veidot programmu no blokiem. Cik vienkārši!
4. **Centrēšanas rīks** – Ja sanāk aizbīdīt ekrānu un pazaudēt blokus, tad nospiežot šo pogu varat iecentrēt visu kodu ekrāna vidū.
5. **Tālummaiņas pogas** – Tuviniet vai attāliniet darba virsmu.

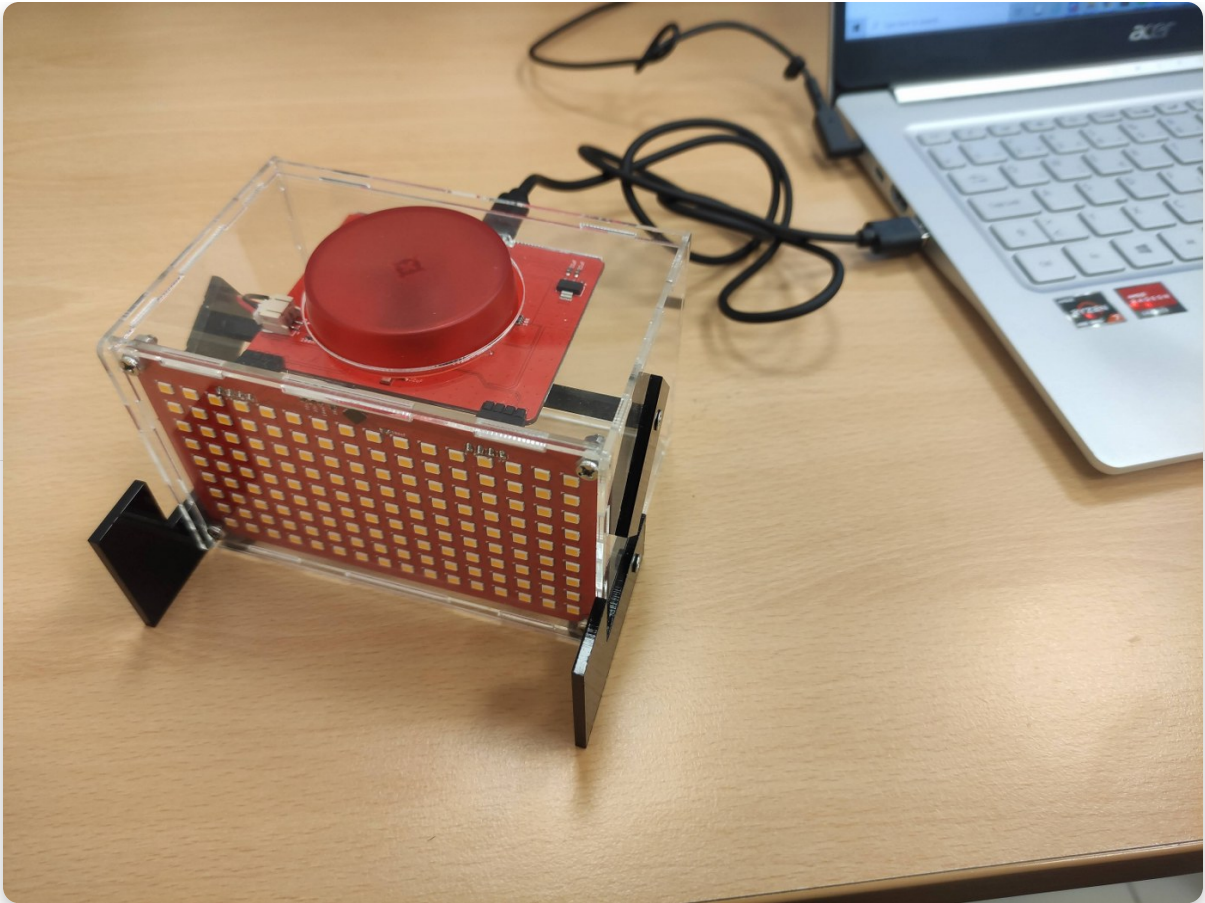
Soli pa solim

Pirmā programma

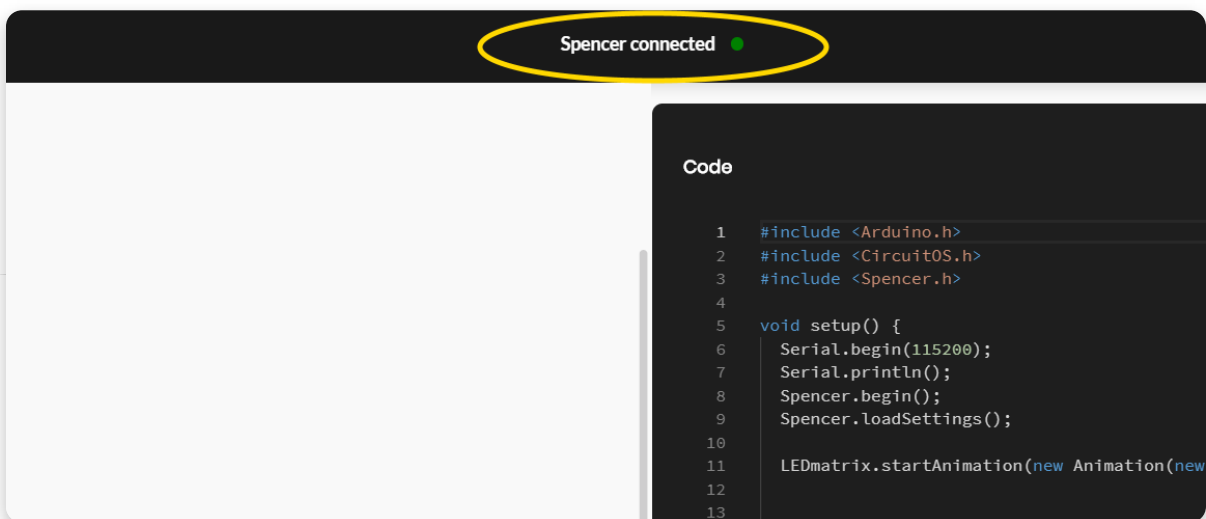
Laiks ķērties klāt pirmās aplikācijas programmēšanai

Pirmkārt, jums jāpieslēdz Spencer pie datora ar USB kabeli.





CircuitBlocks jāuzrāda ka “Spencer connected”



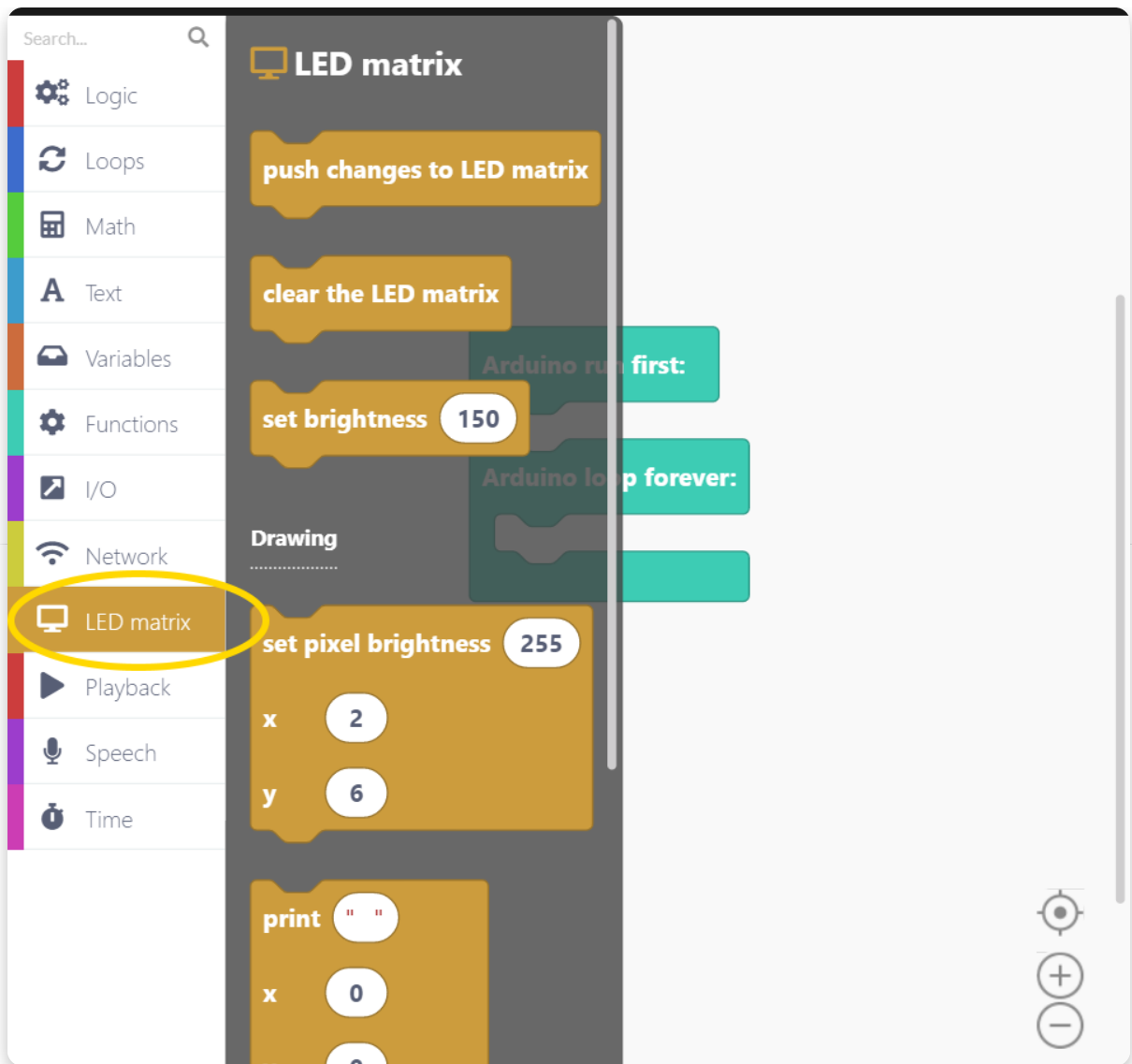
Ja CircuitBlocks neredz jūsu Spencer, lūdzu pārbaudiet vai USB kabelis ir pievienots pareizi un USB ports jūsu datorā strādā.

Ja joprojām dators neredz jūsu Spencer, tad iespējams kaut kas nav kārtībā ar datora draiveriem. Draiveri ir tās mazās programmas kas palīdz datoram komunicēt ar Spencer. Ja par šo tēmu rodas jautājumi, tad sazinaties ar mums – contact@circuitmess.com.

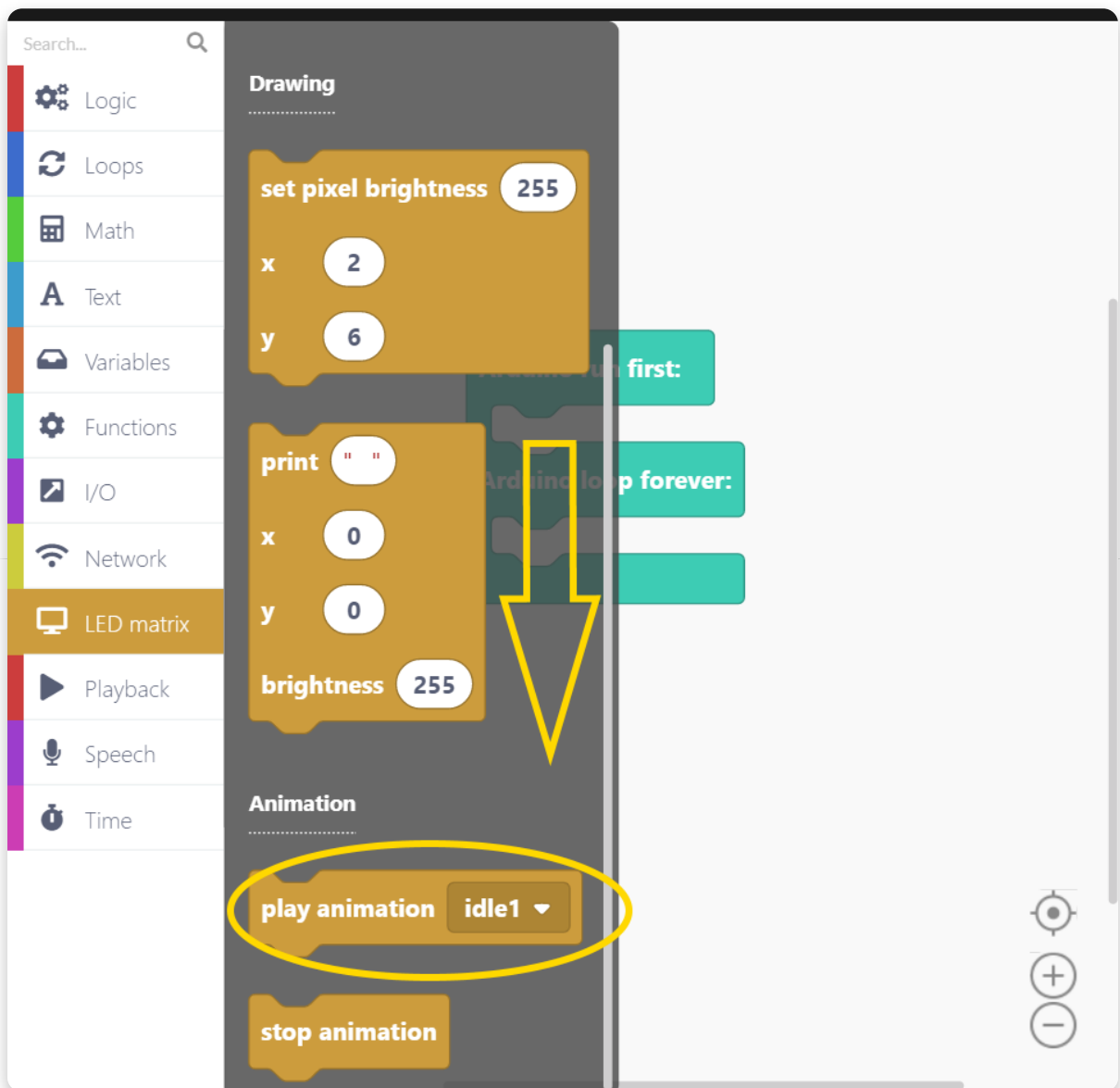
Izveidosim programmatūru lai Spencer parāda animāciju uz displeja

Atrodiet sadaļu ar nosaukumu “LED matrix” kreisajā ekrāna pusē.

Šajā bloku grupā atradīsiet dažādas komandas kas liek Spencer uz displeja parādīt dažādus attēlus.



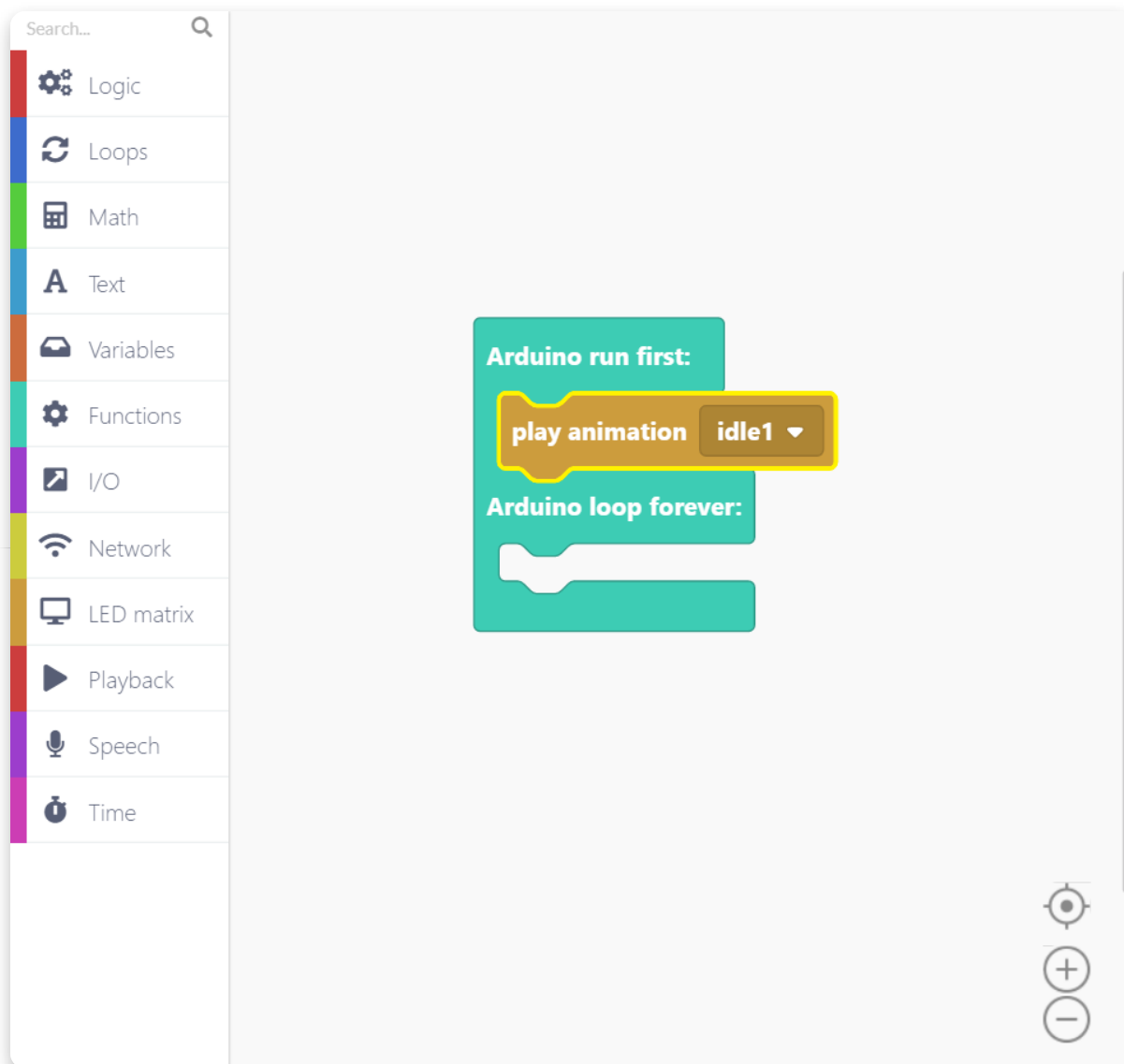
Atrodiet bloku ar nosaukumu "Play animation"



Satveriet bloku ar kursoru un aizbīdi uz jau esošo bloka daļu, pievienojiet tur kur

rakstīts "Arduino run first"

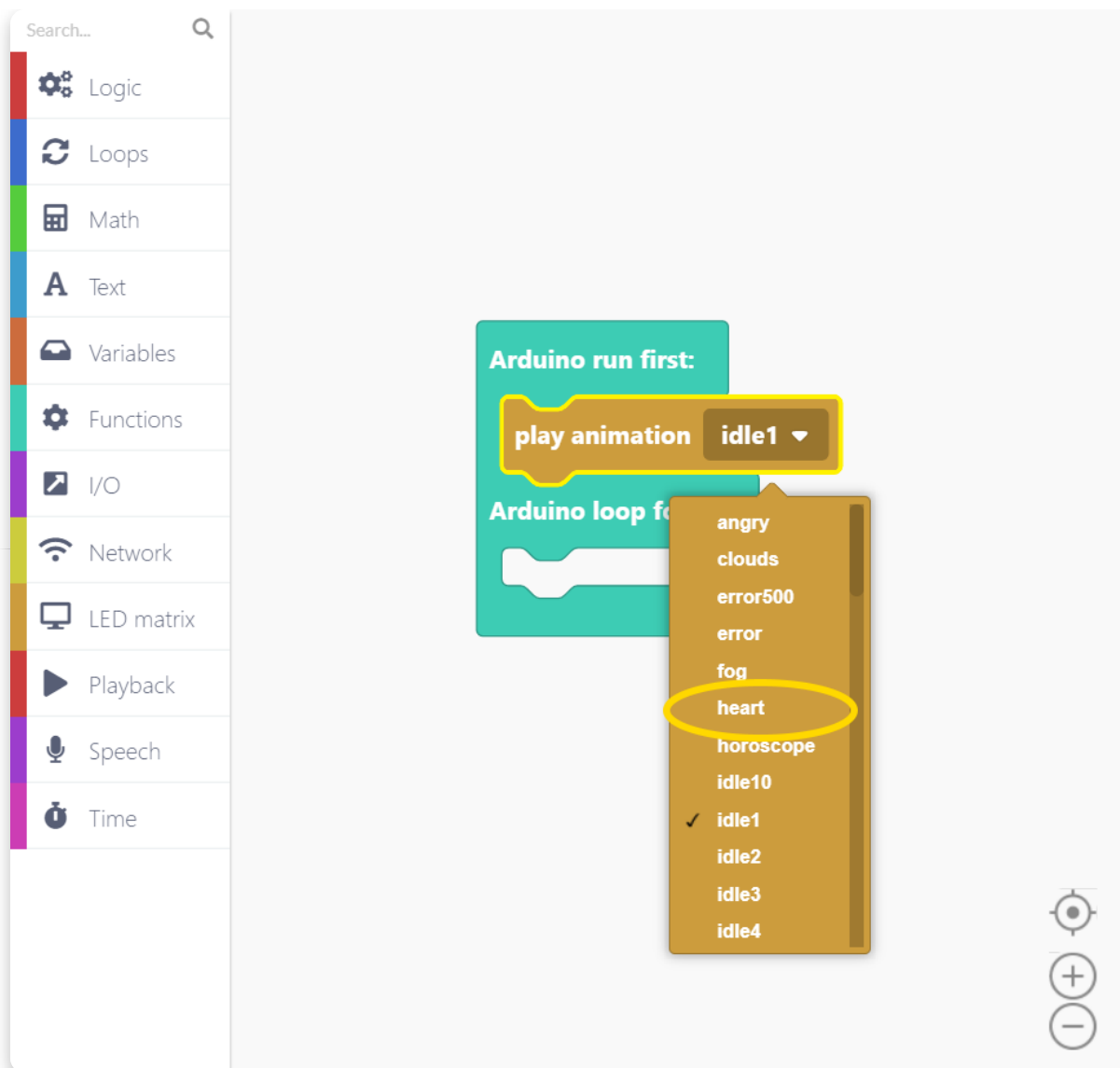
Mēs vēlamies likt Spencer parādīt animāciju tiklīdz tas tiek ieslēgts.



Nospied mazo bultiņu uz "Play animation" bloka un parādīsies visas animācijas kas ir saglabātas Spencer iebūvētajā flash atmiņā. Pagaidām jaunas animācijas pievienot nav iespējams, taču mēs pie tā strādājam, lai ar jaunu CircuitBlocks atjauninājumu tas būtu iespējams.

Taču jums arī ir iespēja iedarbināt katru atsevišķu pikseli un LED paneļa, un to tēmu apskatīsim vēlāk.

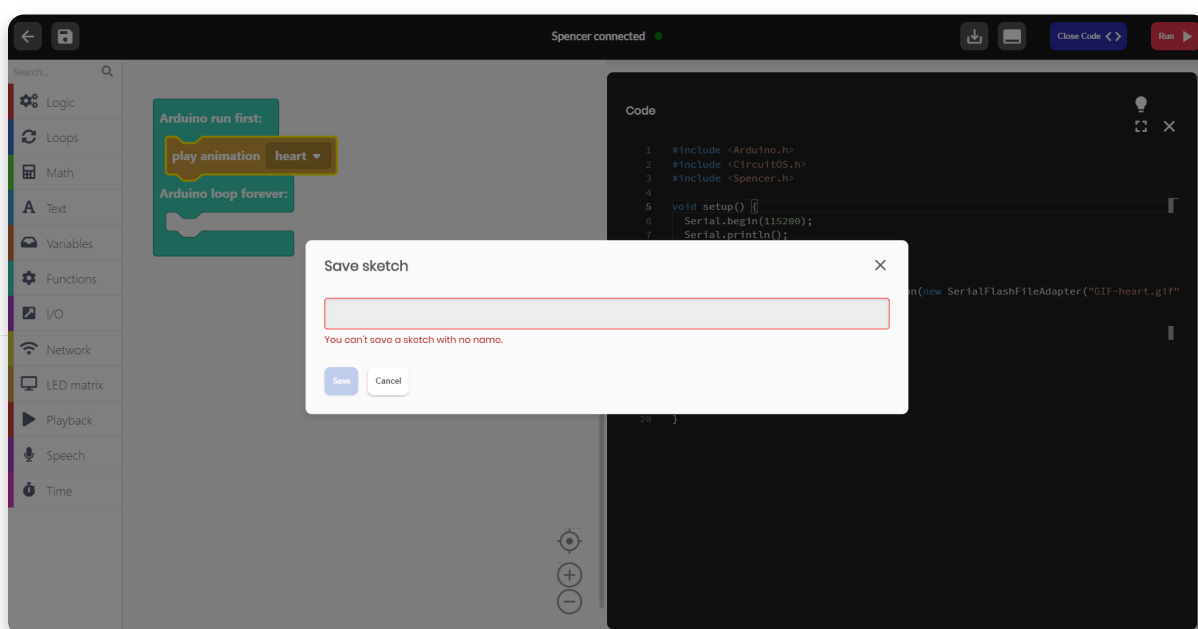
Ak jā, turpinam apmācību. Izvēlies "heart" animāciju.



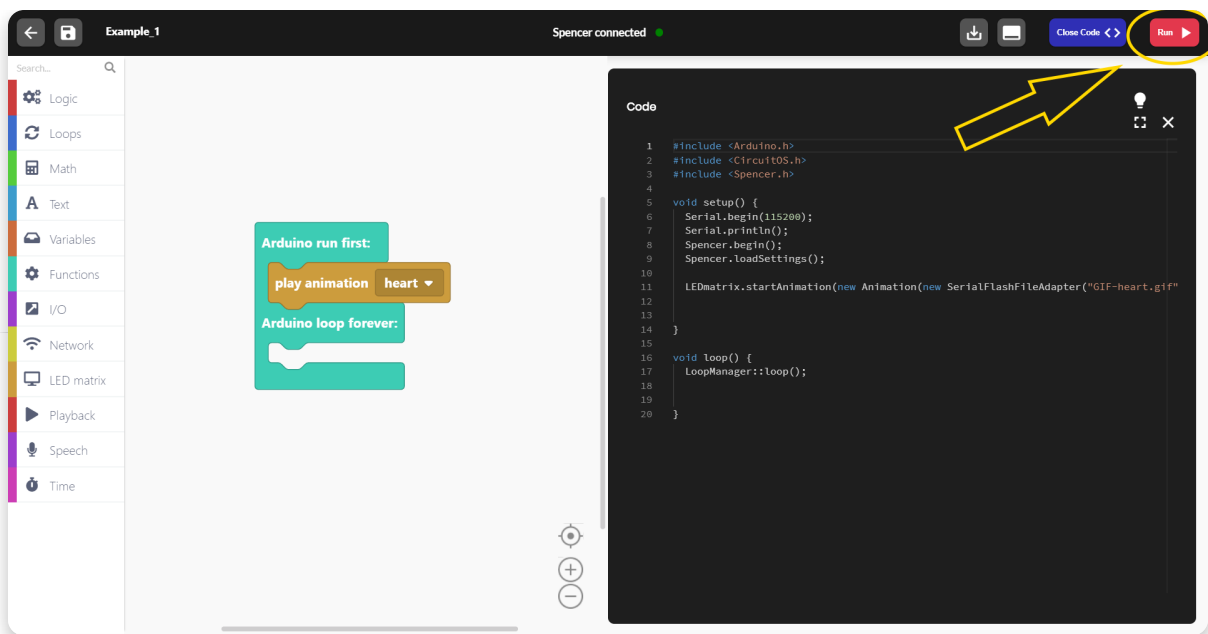
Lieliski! Mūsu pirmā programma ir gatava !

Šis tikko uztaisītais kods liks Spencer parādīt animāciju uz LED matricas. Tiklīdz Spencer tiks ieslēgts, parādīsies animācija un tiks atskaņota bezgalīgi.

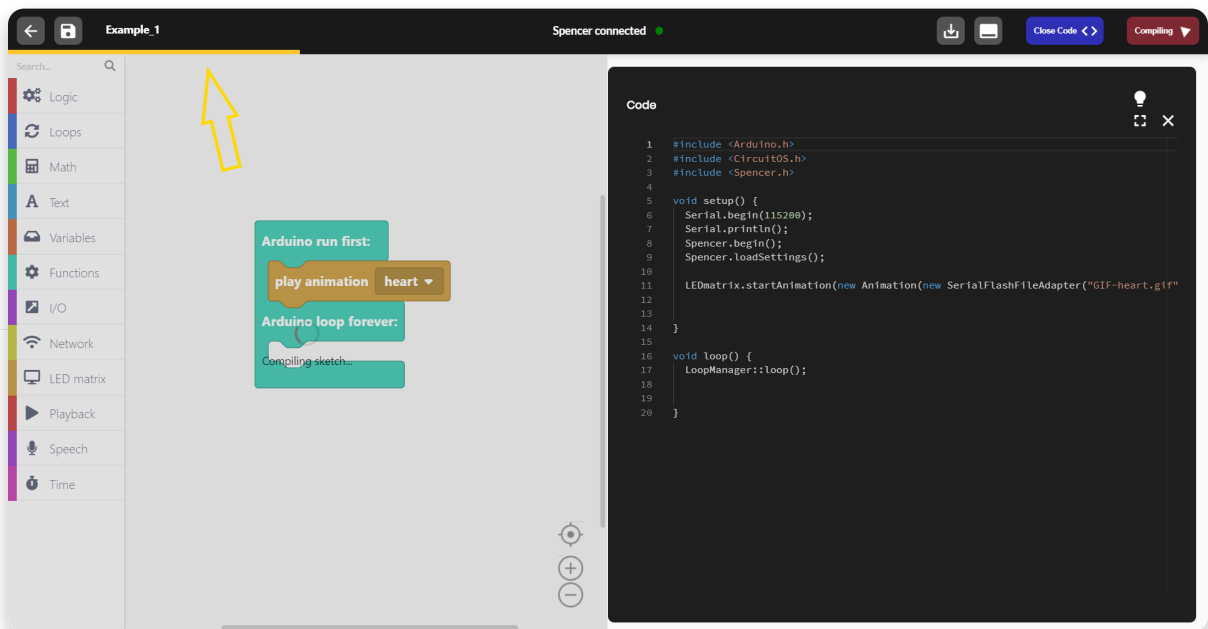
Pirms turpinām, nospiedīsim SAVE pogu, un nosauksim projektu.



Kad tas izdarīts, sameklē sarkano pogu RUN , un nospied to



Dzeltena ielādes josla parādīsies ekrāna augšdaļā.



Jūsu programma šobrīd ir sastādīšanas procesā

Koda kompilēšana cilvēkam saprotama koda (piemēram krāsainie bloki ko redzat) pārveidošana par iekārtai saprotamu kodu (vieninieki un nullītes ko saprot Spencer)

Par cik šī ir pirmā reize kad tiek kompilēts programmas kods priekš Spencer, tas var aizņemt pāris minūtes. Tas notiek tāpēc, ka CircuitBlocks ir jāapkopo visas svarīgās koda daļas, kas nepieciešamas jūsu Spencer, un jā saglabā tas datorā. Pirmo reizi apkopojot, tas tiks saglabāts jūsu datorā, un visas nākamās programmas tiks apkopotas daudz ātrāk (yay!).



Ja jūsu kods tika veiksmīgi apkopots un augšupielādēts, jums vajadzētu redzēt sirds animāciju, kas tiek atskaņota jūsu Spencer.

Vai viss kārtībā? Lieliski, ejam tālāk.

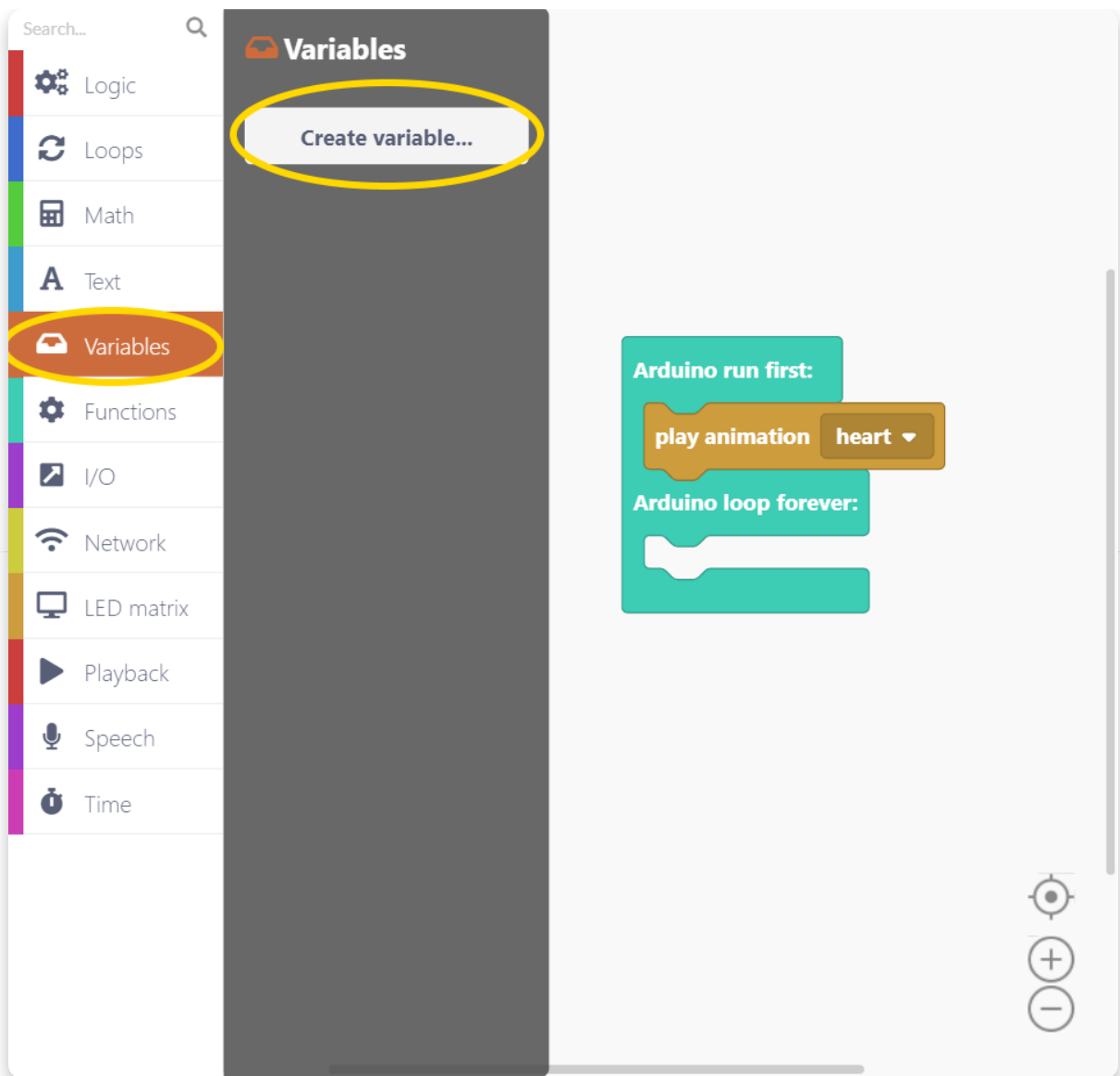
Kaut kas nedarbojas? Lūdzu, nospiediet CircuitBlocks mājas lapā ziņojumu Sūtīt kļūdu, sazinieties ar mums pa e-pastu contact@circuitmess.com un sniedziet mums savu kļūdas ziņojuma ID.

Izmēģināsim mainīgos

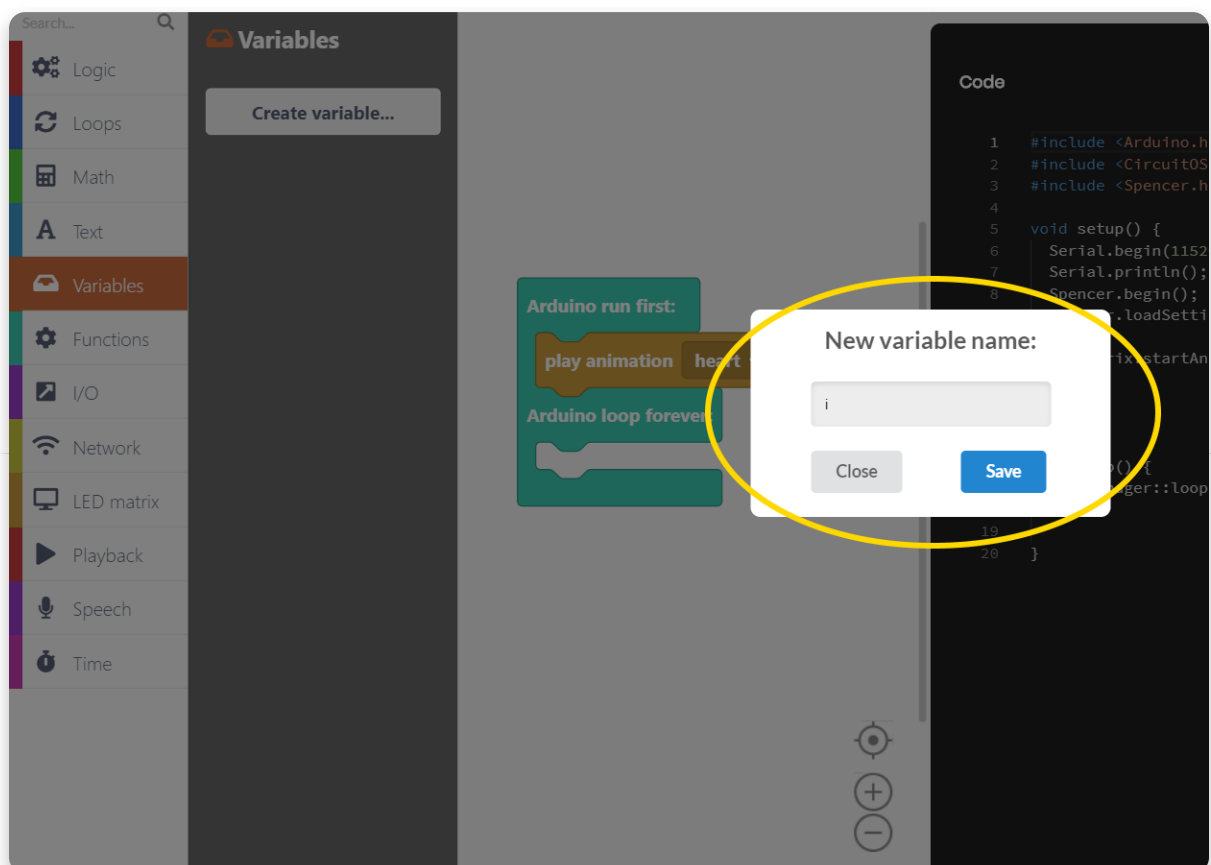
Nākamajā solī dodamies mazliet tālāk un izmantojam **dažus mainīgos**.

Datorprogrammēšanā mainīgais ir glabāšanas vieta, kurā ir vērtība. Katram mainīgajam ir noteikts nosaukums. Varat saglabāt un mainīt mainīgā vērtību.

Pirmkārt, izveidosim mainīgo. Atrodiet sadaļu ar nosaukumu "Variables" un nospiediet pogu "Create variable ...".



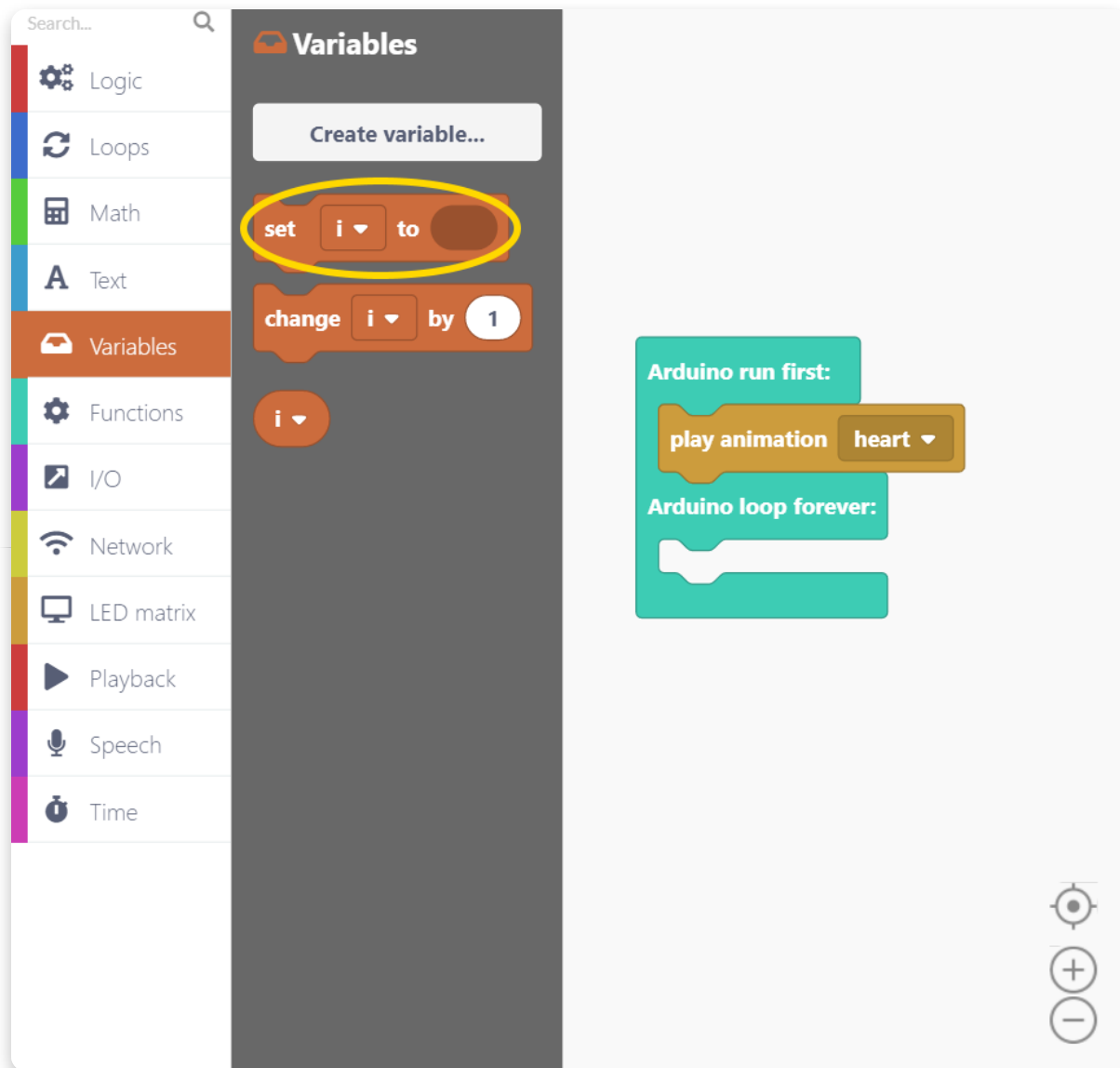
Jums mainīgajam jāpiešķir nosaukums. Šajā gadījumā nosauksim to par "i".



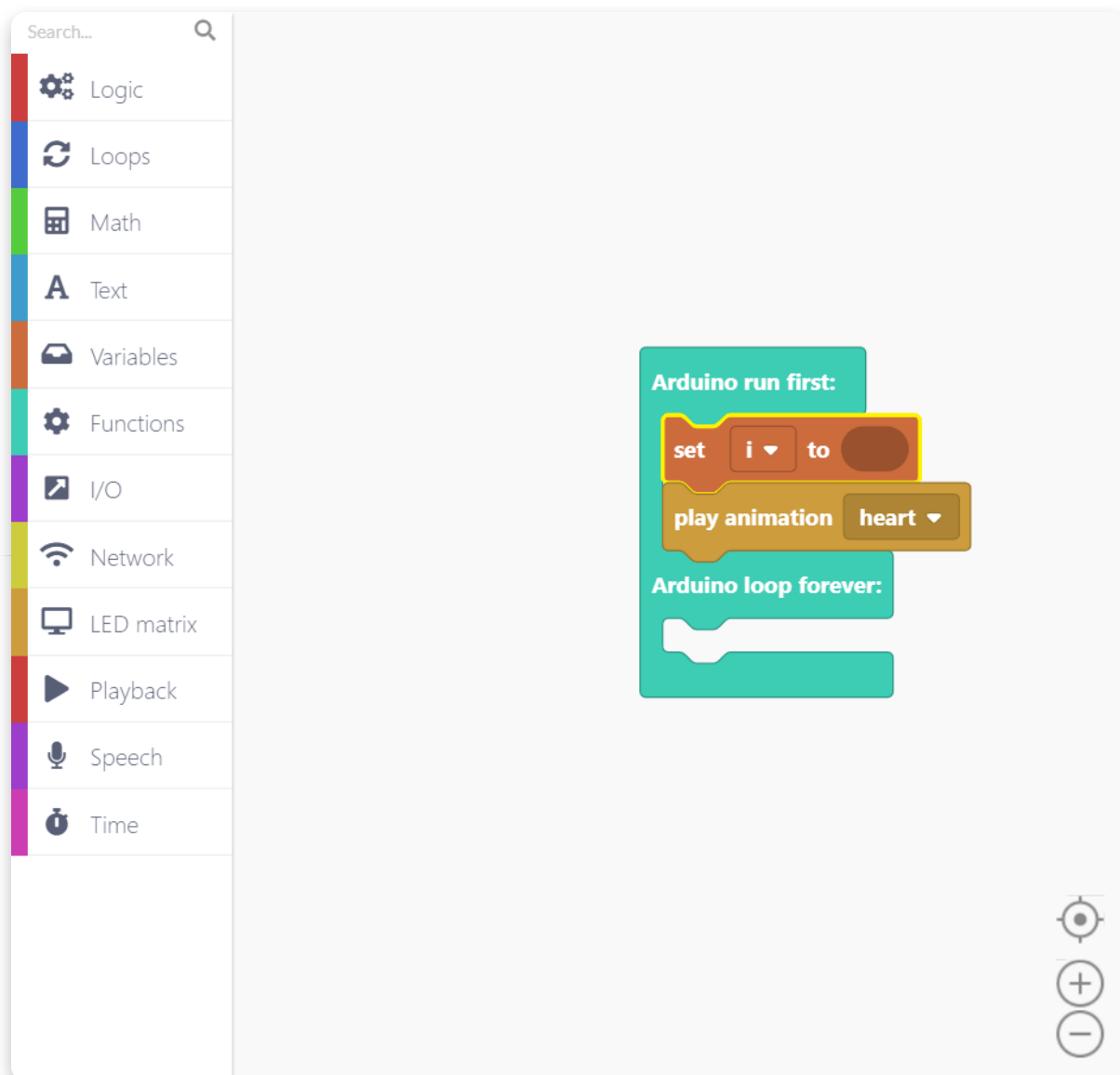
Tagad mums ir mainīgais. Lieliski!

Kad mēs izveidojam mainīgo, tas nav definēts – tam nav vērtības. Startējot mūsu datorprogrammu, katram mainīgajam ir jāiestata vērtība. Tāpēc jums būs

nepieciešams bloks "set variable".

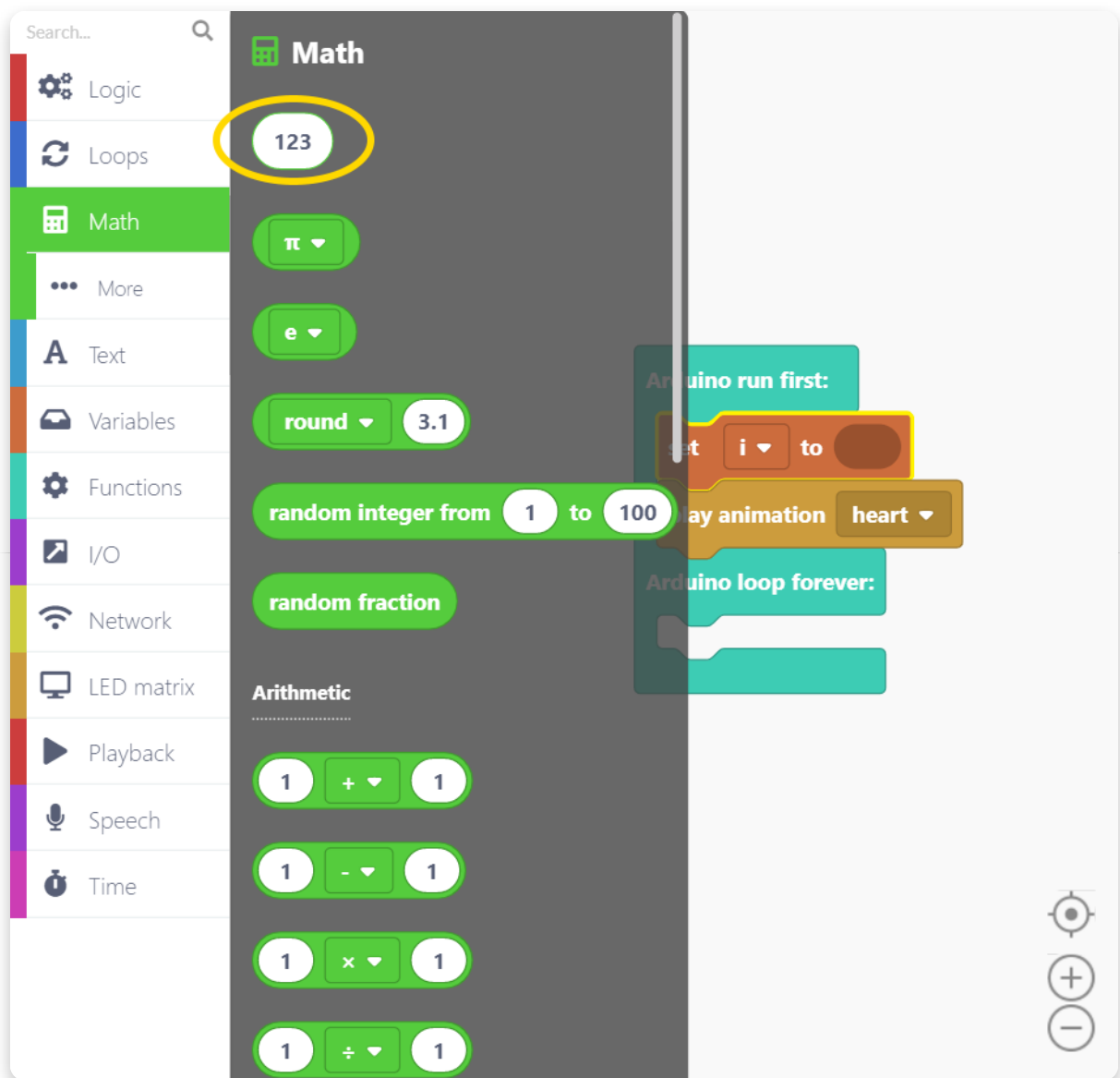


Šo bloku ievietojat "Arduino run first" vietā.

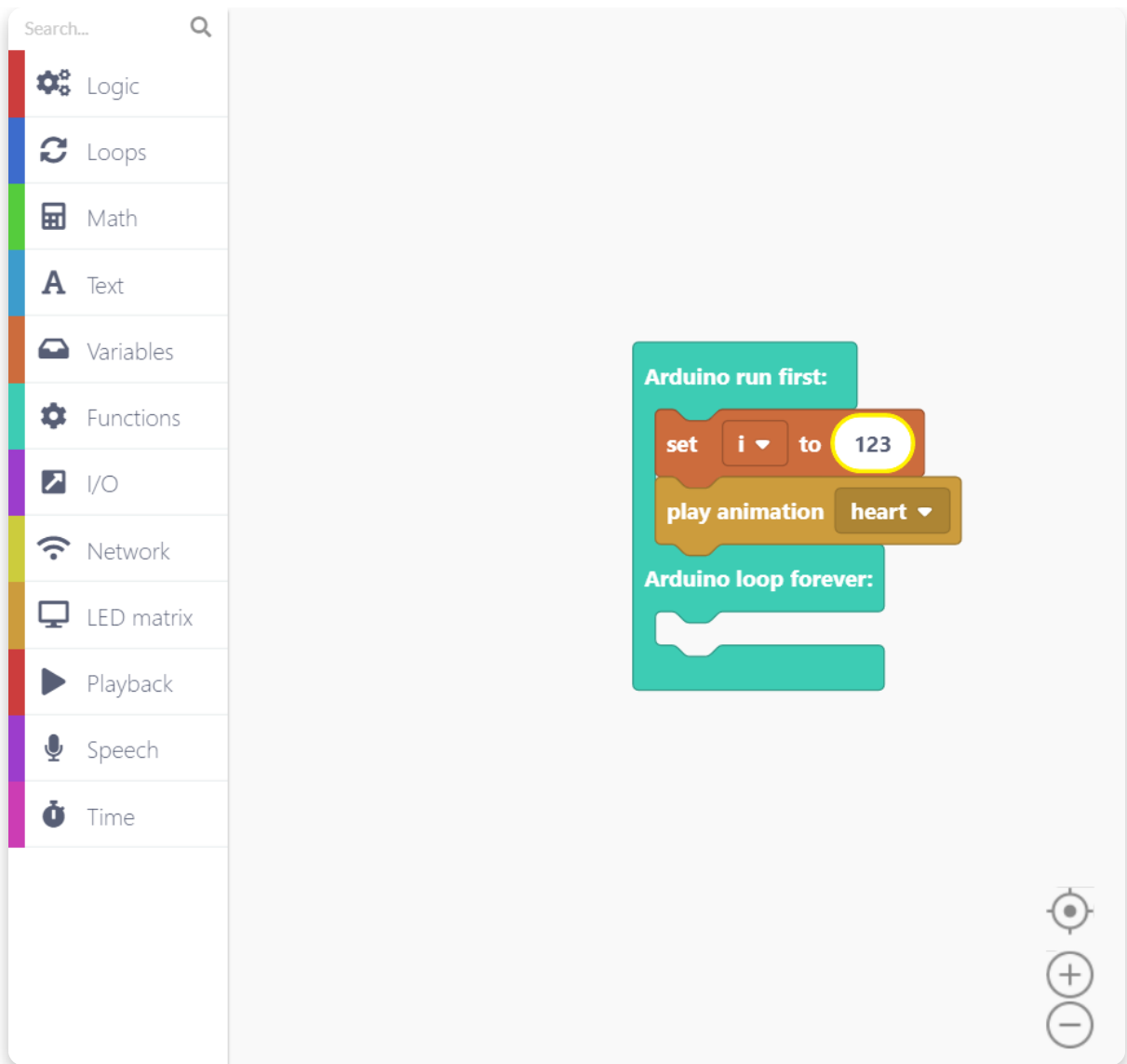


Tagad jums ir jādefinē vērtība, kurai mēs vēlamies iestatīt mainīgo.

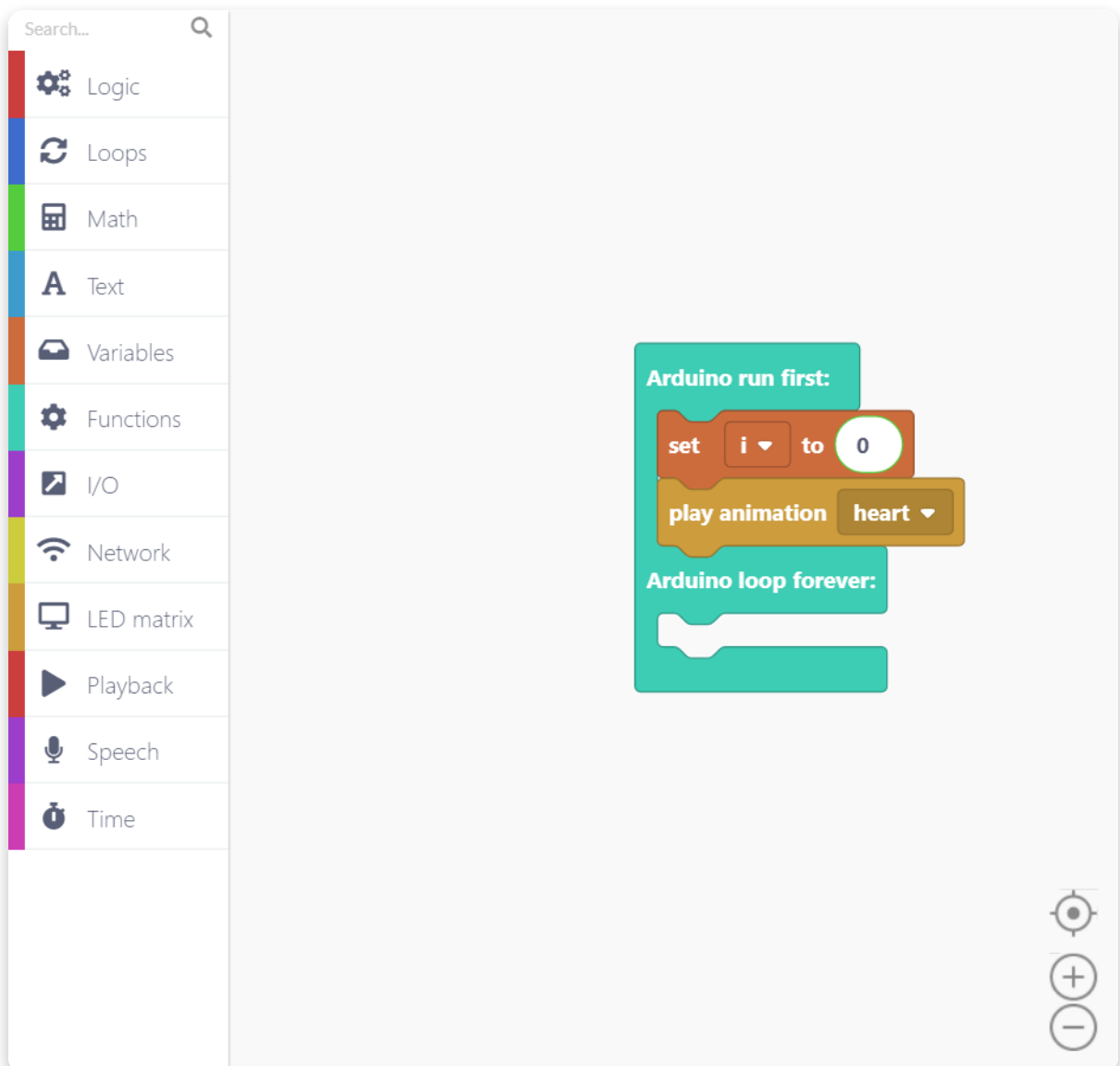
Sadaļā "Math" atrodiēt šo bloku ar nosaukumu "123". Šis bloks ir skaitlisko vērtību bloks, un, ievadot vajadzīgo skaitlisko vērtību, nometiet to zīmēšanas apgabalā.



Novietojat bloku šeit:



Tagad noklikšķiniet uz bloka un ierakstiet vērtību. Iestatiet bloka skaitlisko vērtību uz 0. To var izdarīt, vienkārši ievadot tastatūras skaitli 0 (nulle).

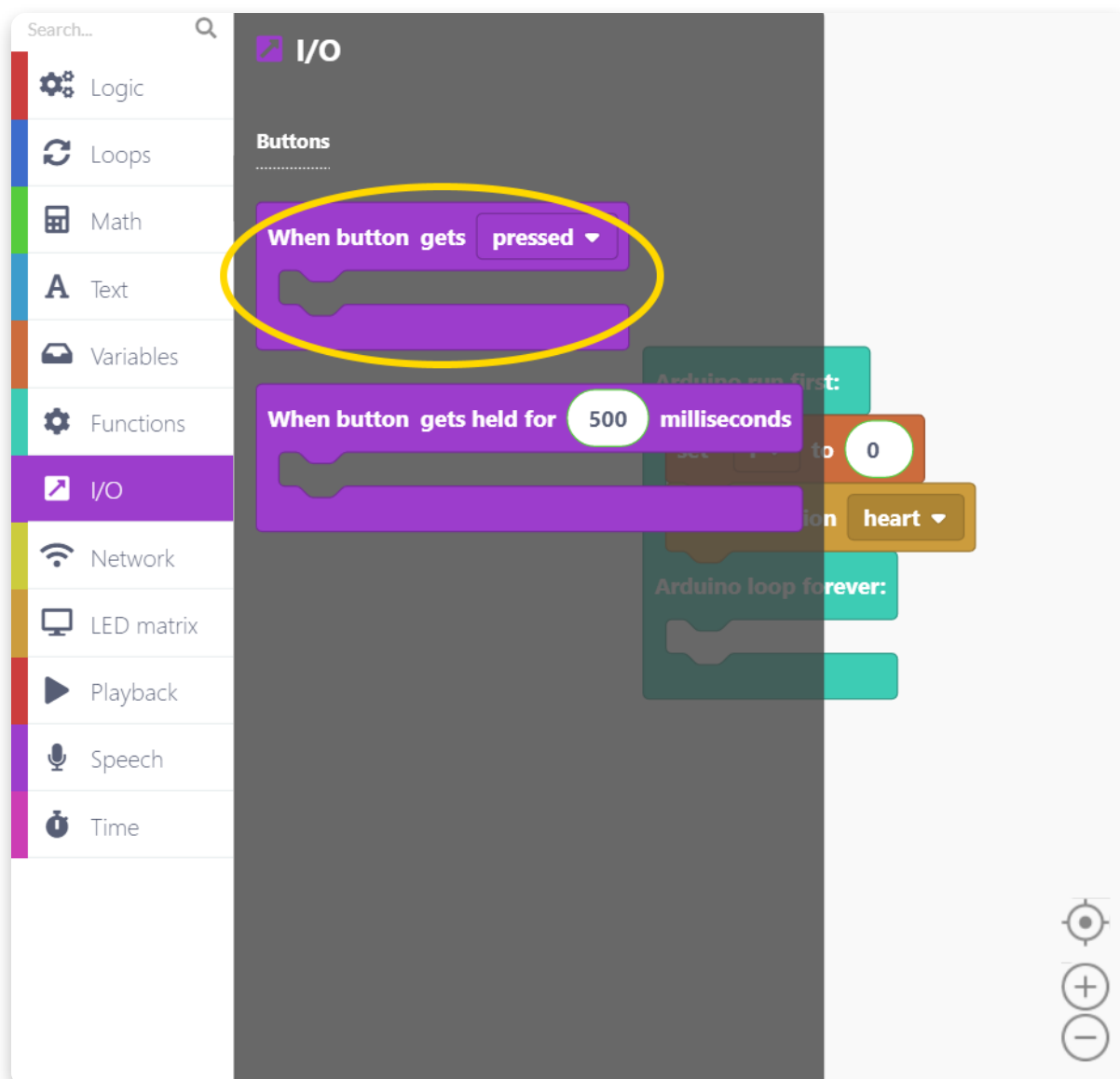


Tagad, kad mums ir izveidots mainīgais un iestatīts uz nulli, mainīsim mainīgo, kad tiks aktivizēts noteikts notikums.

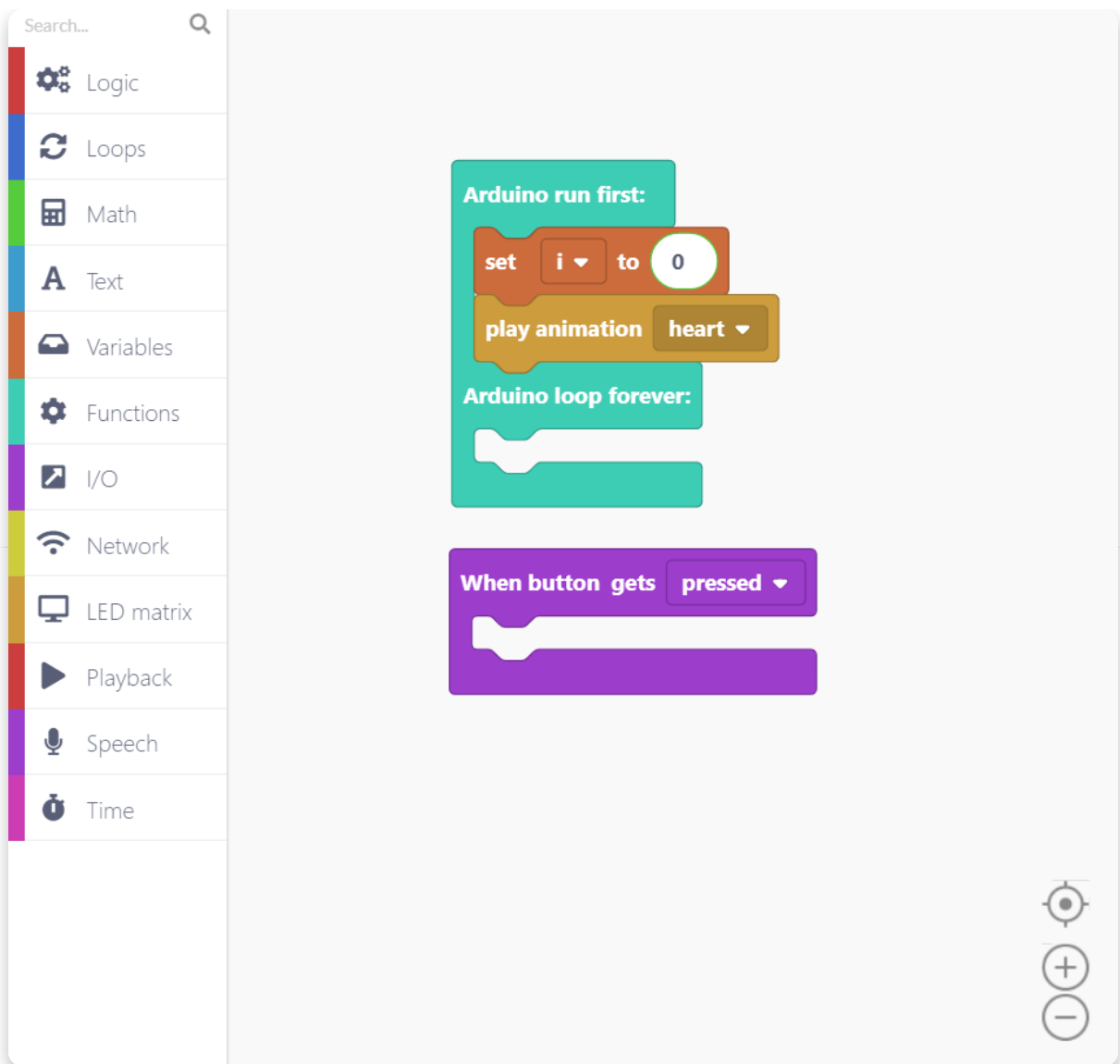
Mēs mainīsim mainīgo, kad jūs uzsitīsiet savu Spensera lielo sarkano pogu.

Par laimi, mums ir īpašs bloks, kas definēts Spenser sarkanās pogas piesišānai, un tas atrodas I / O sadaļā. I / O nozīmē "Ieeja / Izvade". Spenser poga ir tā sauktā ievades ierīce, jo tā iedarbina Spencer datoru uz elektrisko impulsu. Spensera displejs ir izvades ierīces piemērs, jo Spensers sūta signālus, lai parādītu informāciju.

Jums jāatrod šis violetais bloks ar nosaukumu "When button gets pressed".

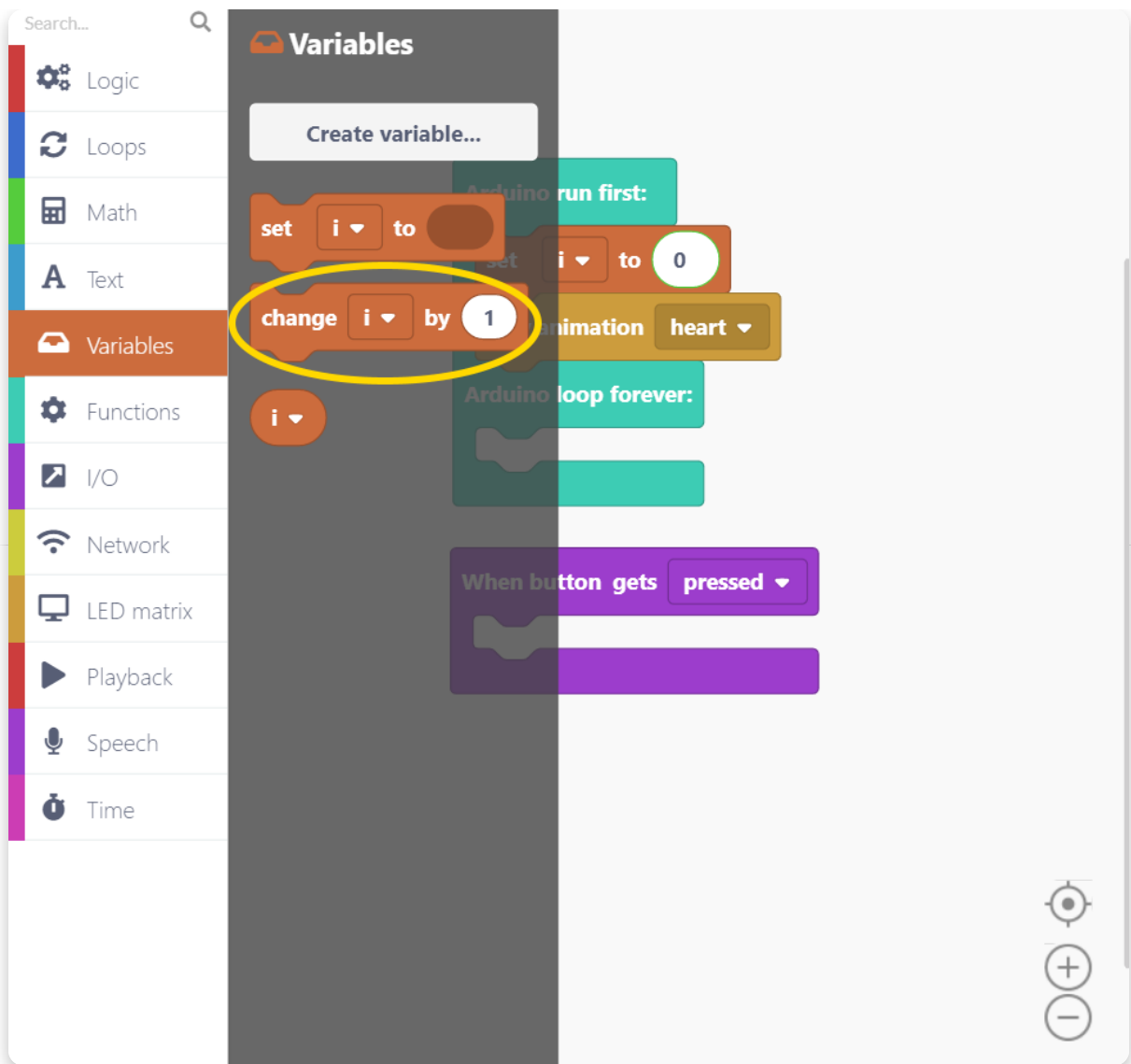


Novieto bloku programmatūras veidošanas logā



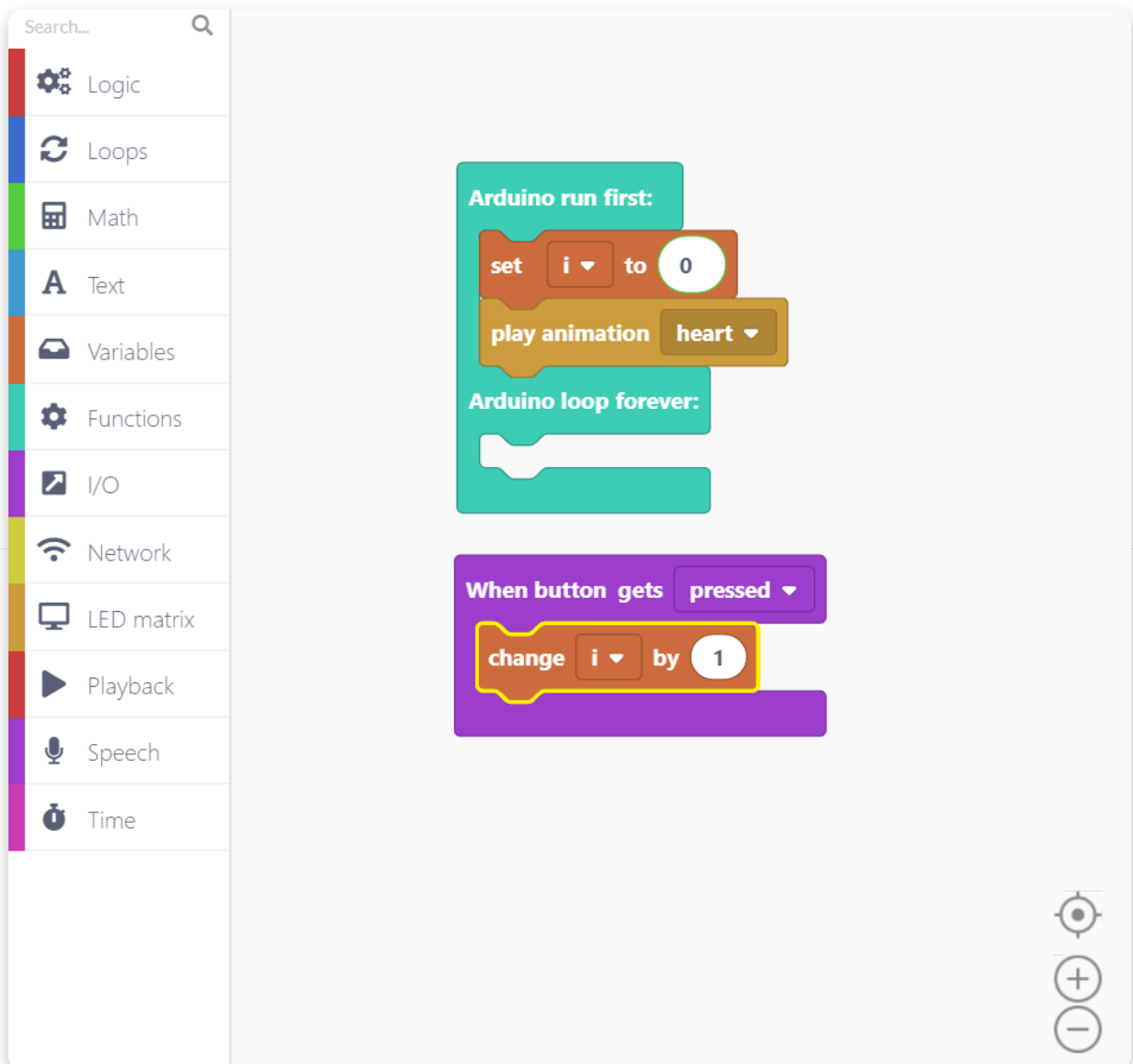
Tagad palielināsim mainīgā vērtību par vienu vienību katru reizi kad tiek piespiesta spencer poga.

To var panākt šādi:



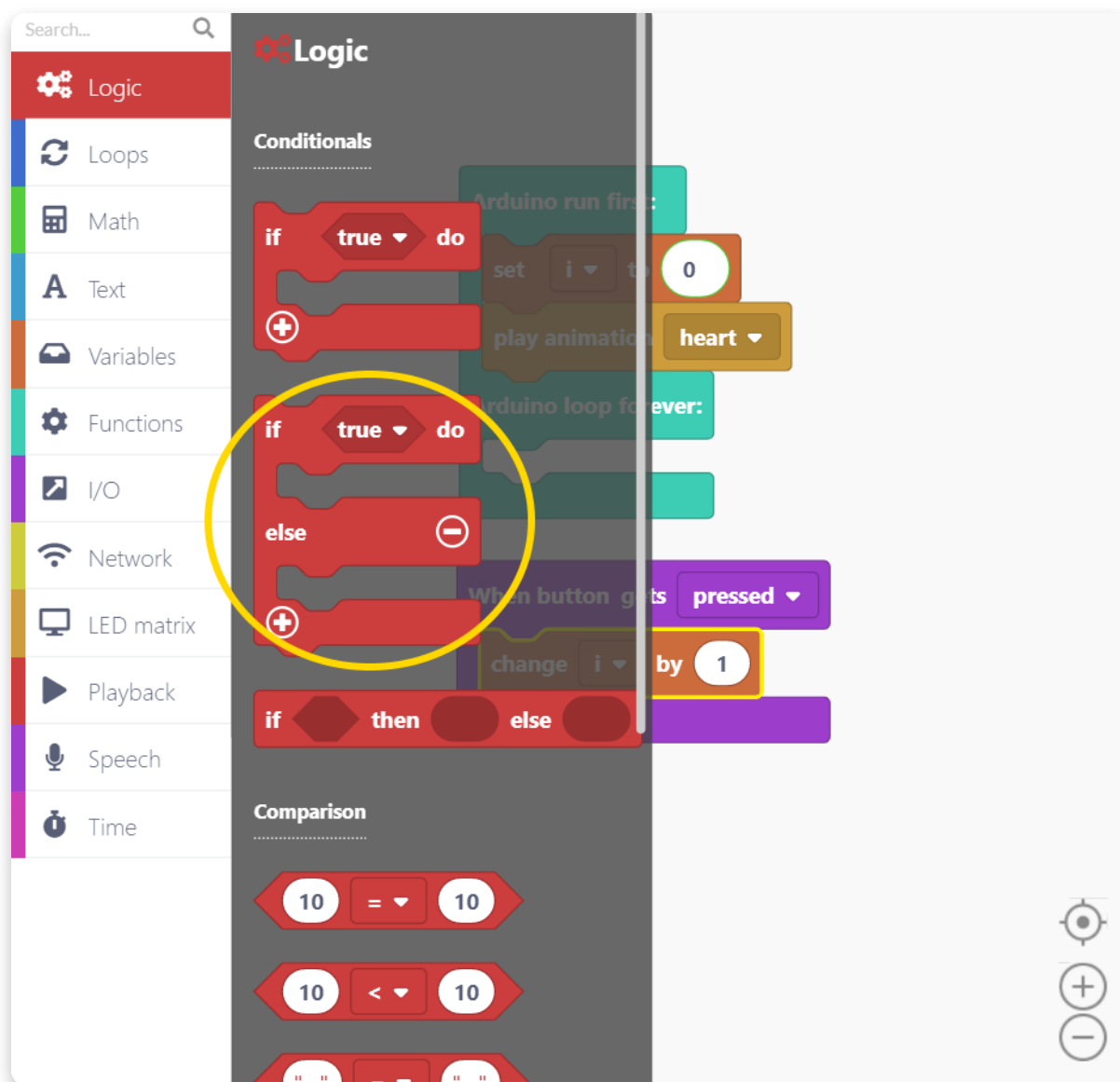
random image

Novieto šo bloku kā redzams attēlā:



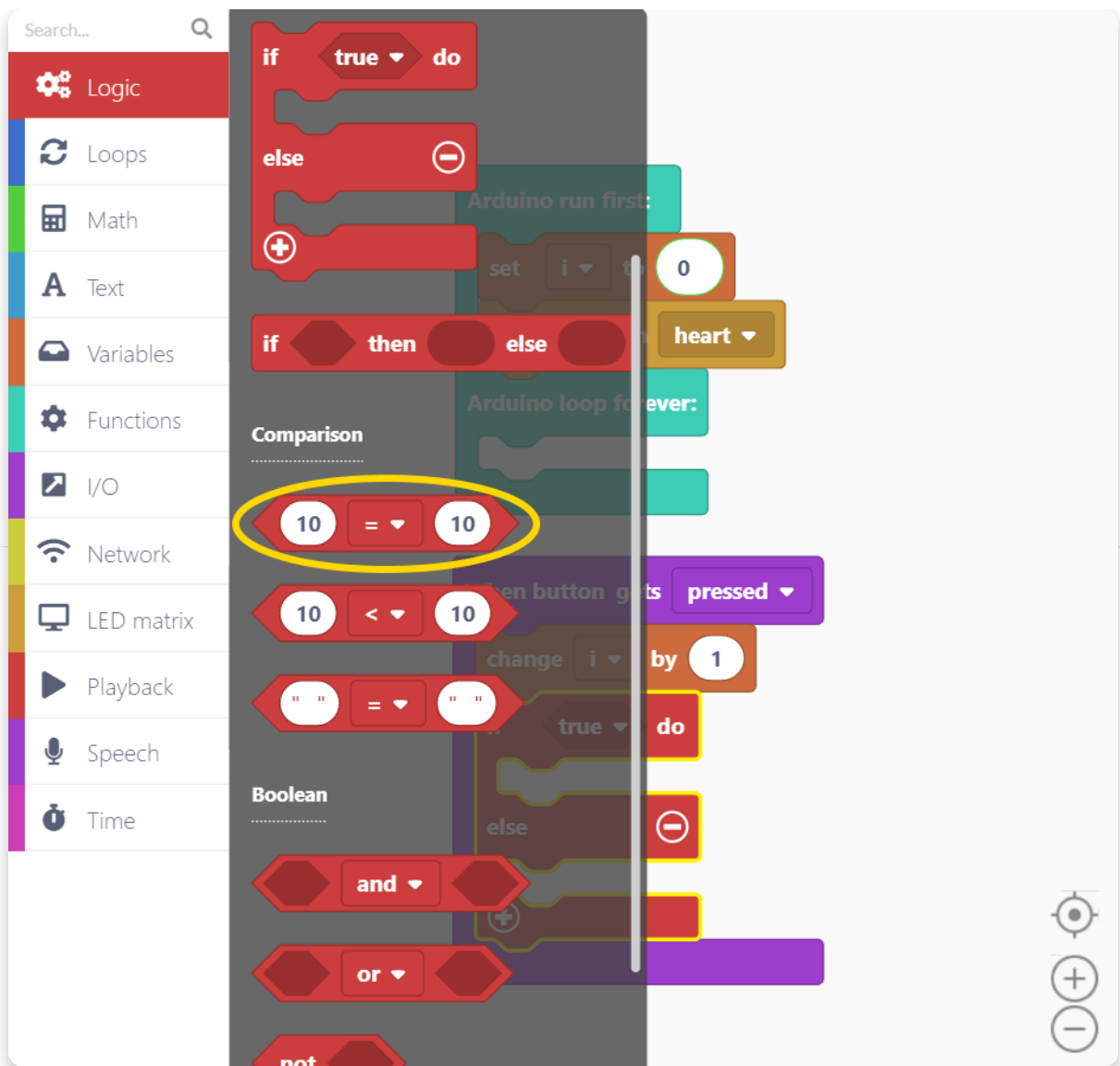
Tagad darīsim kaut ko interesantu - mēs izmantosim loģisku bloku IF-THEN-ELSE, lai Spencera sejā parādītu dažādas animācijas atkarībā no mainīgā mainīgā lieluma.

Sadaļā "Logic" atrodiēt šo sarkano bloku:

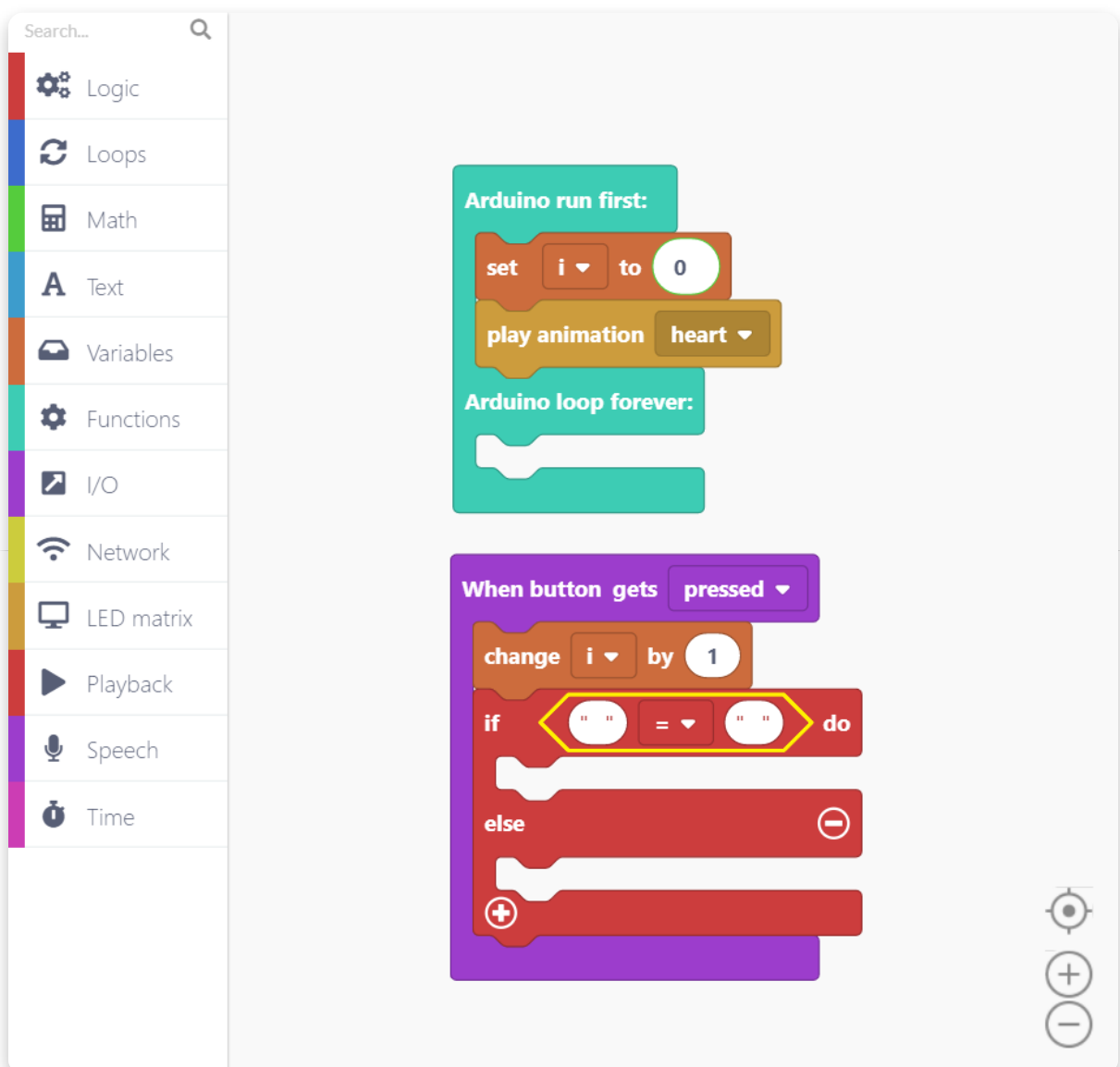


Novietojiet šo bloku violetajā blokā "When button gets pressed block".

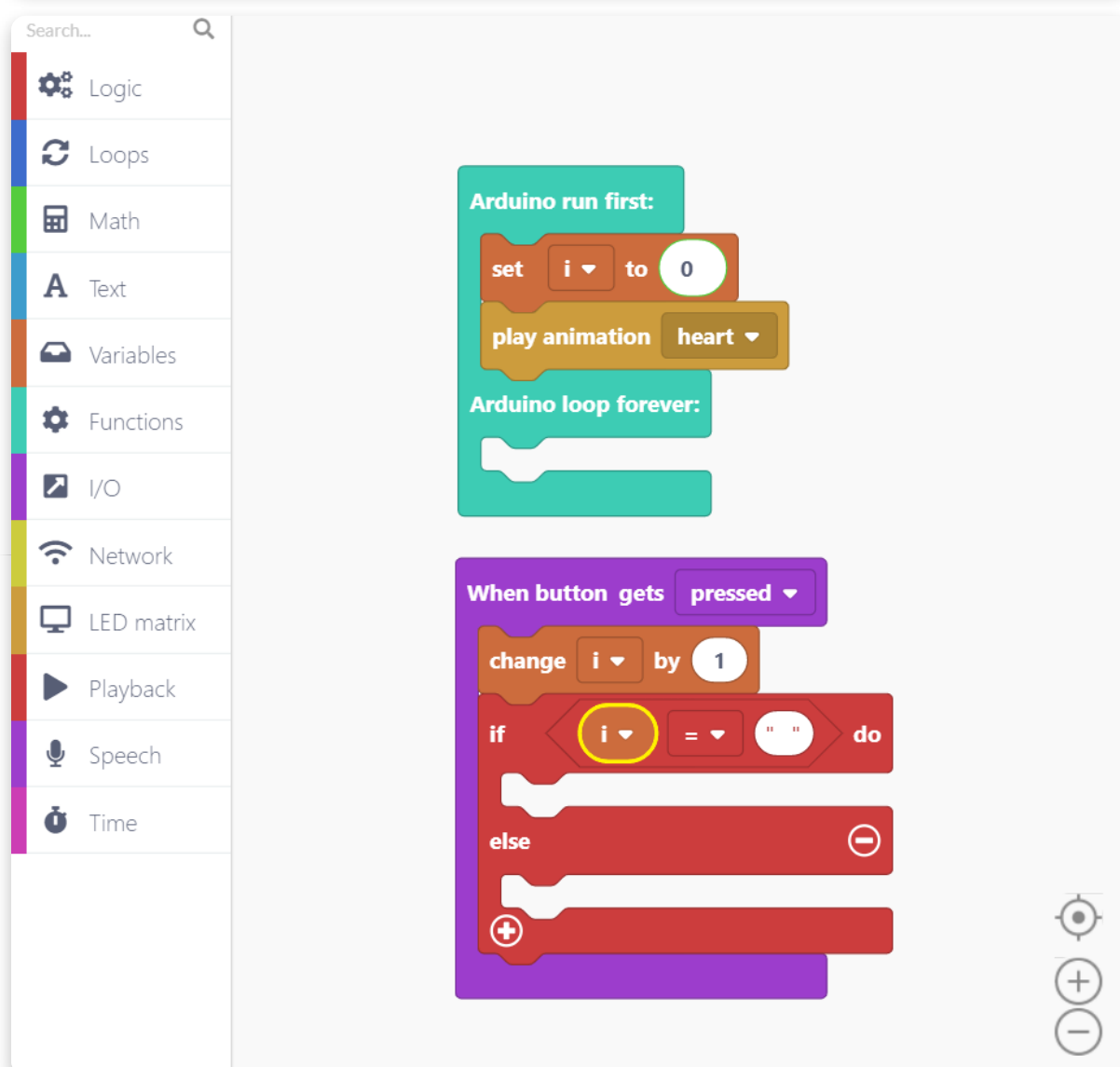
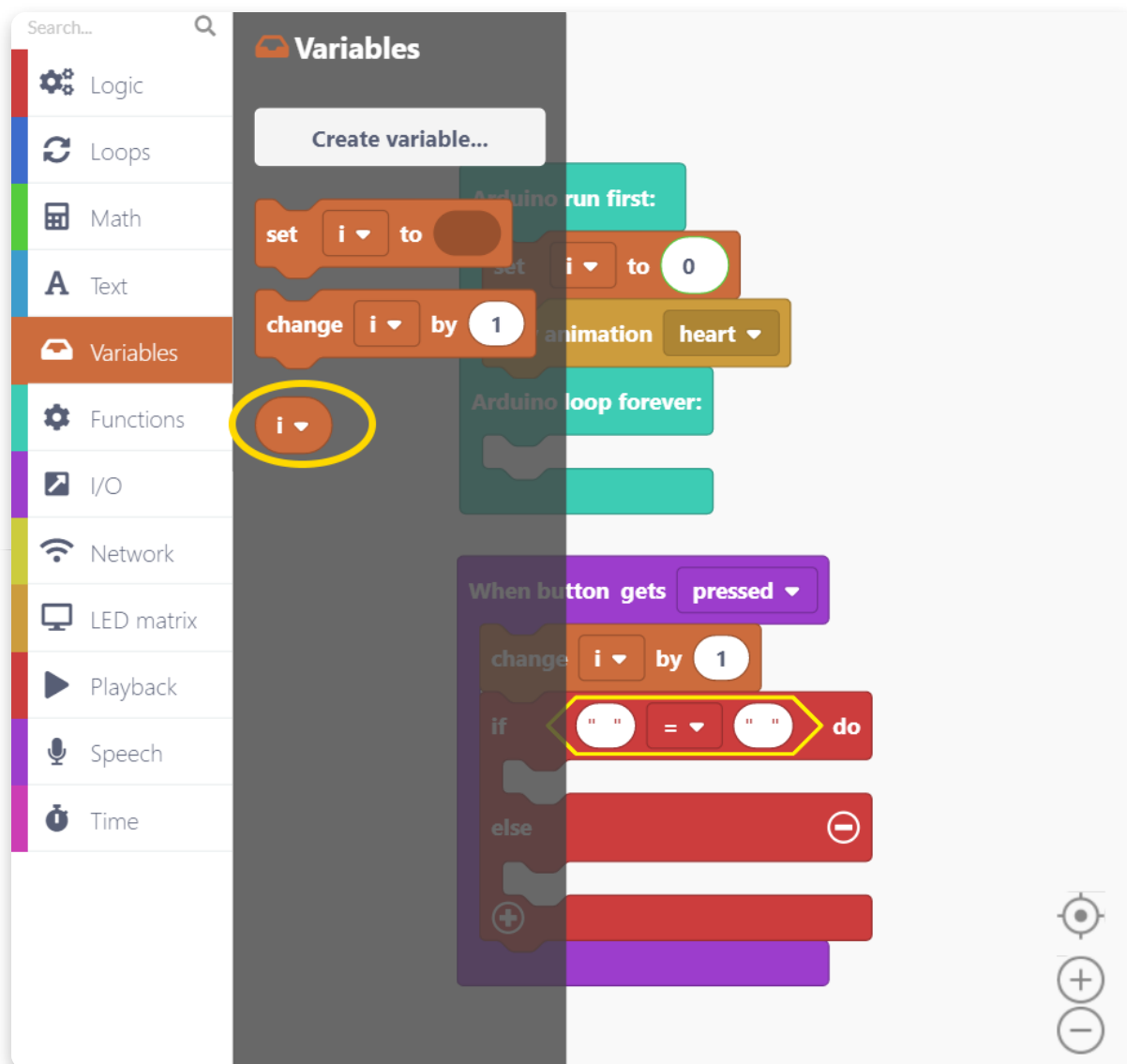
Tālāk jums būs nepieciešams salīdzināšanas bloks. Šo bloku parasti izmanto, lai salīdzinātu mainīgā vērtību ar fiksētu vērtību (t.i., redzēsīm, vai mainīgais i ir vienāds ar skaitli 1).



Novietojiet divainas formas salīdzināšanas bloku loģiskajā blokā IF-THEN-ELSE šādi:

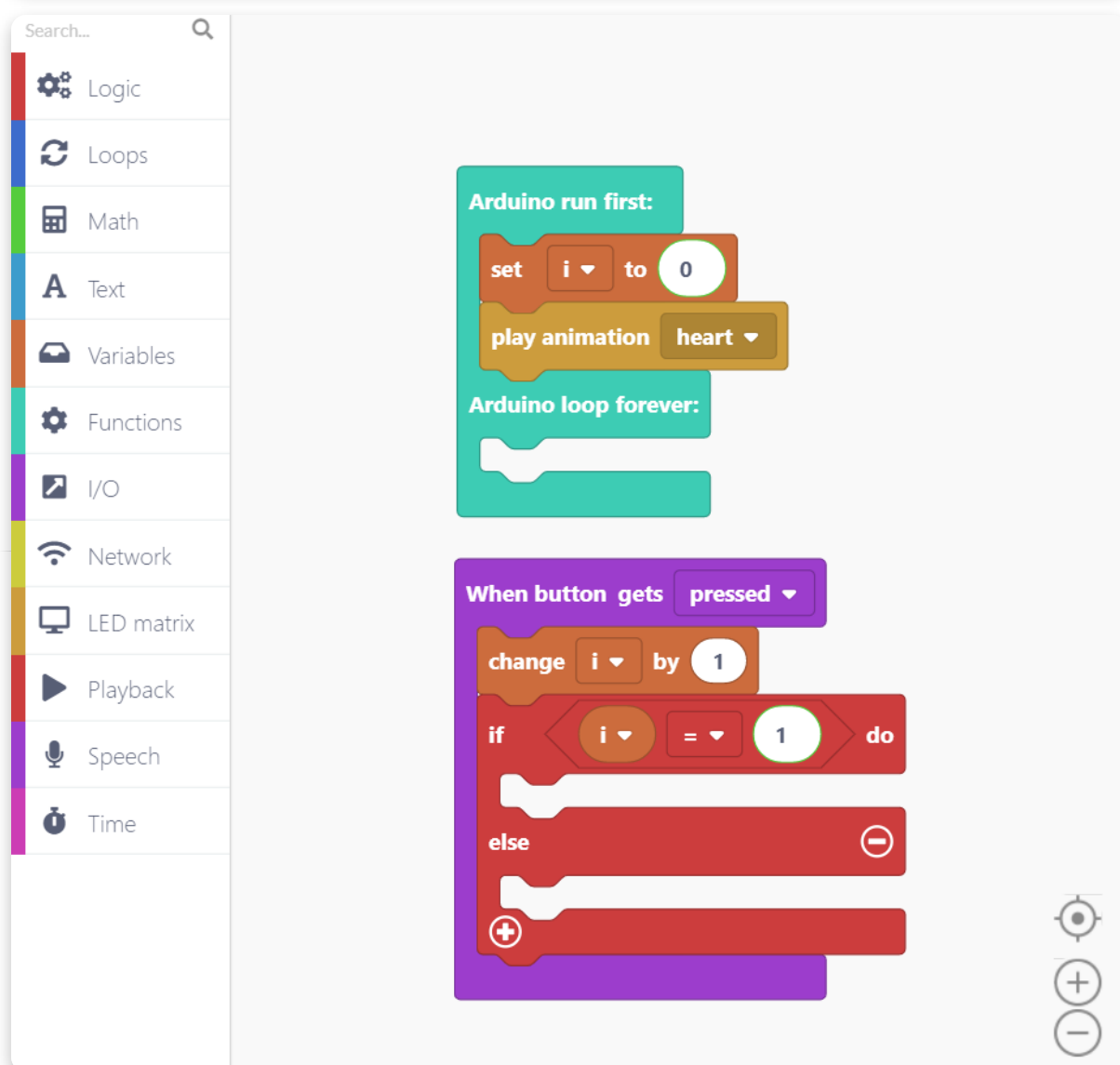
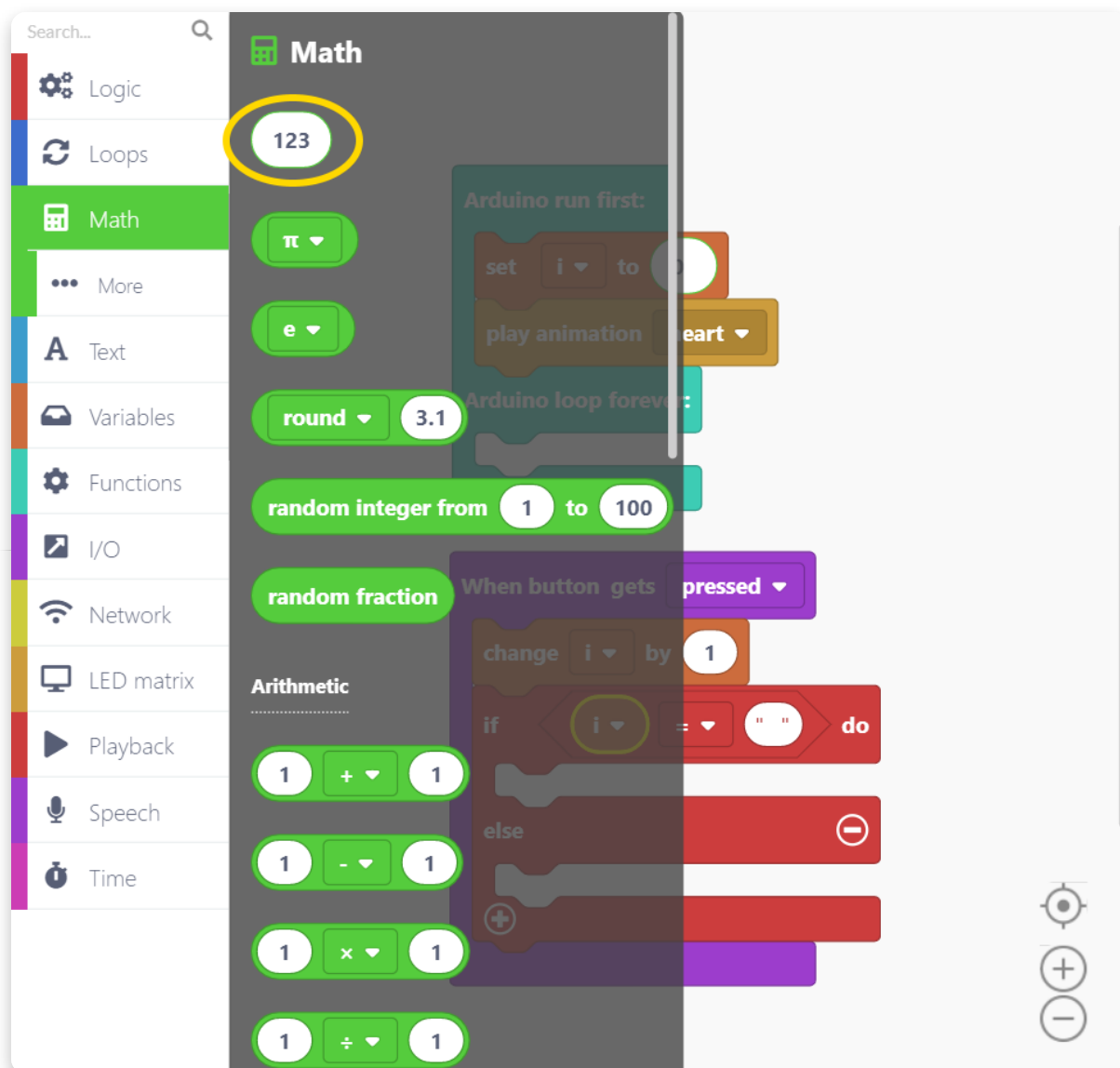


Kreisajā pusē vienādības blokā izvēlaties iepriekš sagatavoto mainīgo i.



Paņemiet skaitlisko vērtību bloku no matemātikas sadaļas un novietojiet to

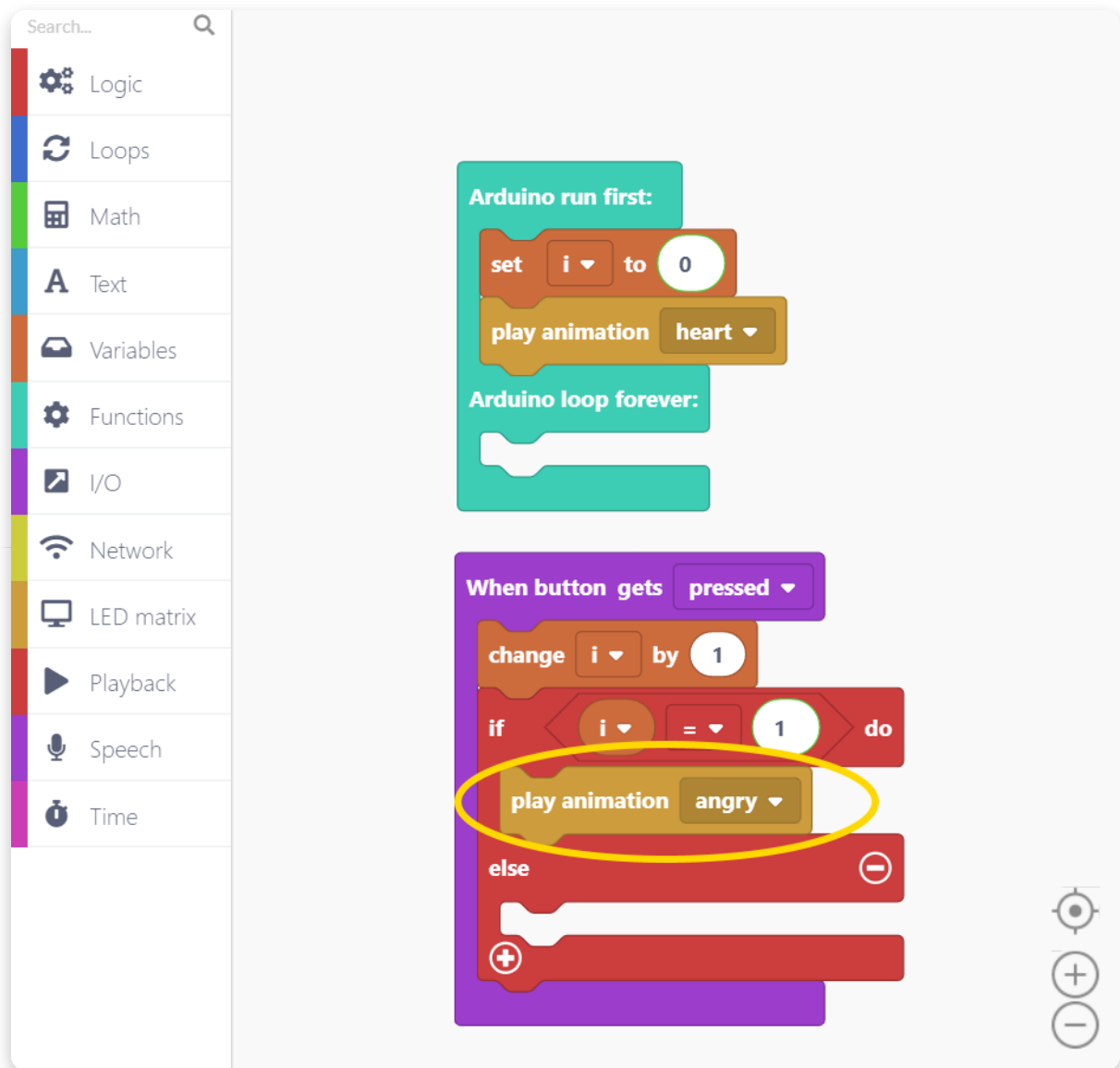
salīdzināšanas bloka labajā pusē šādi:



Ierakstiet blokā skaitli 1.

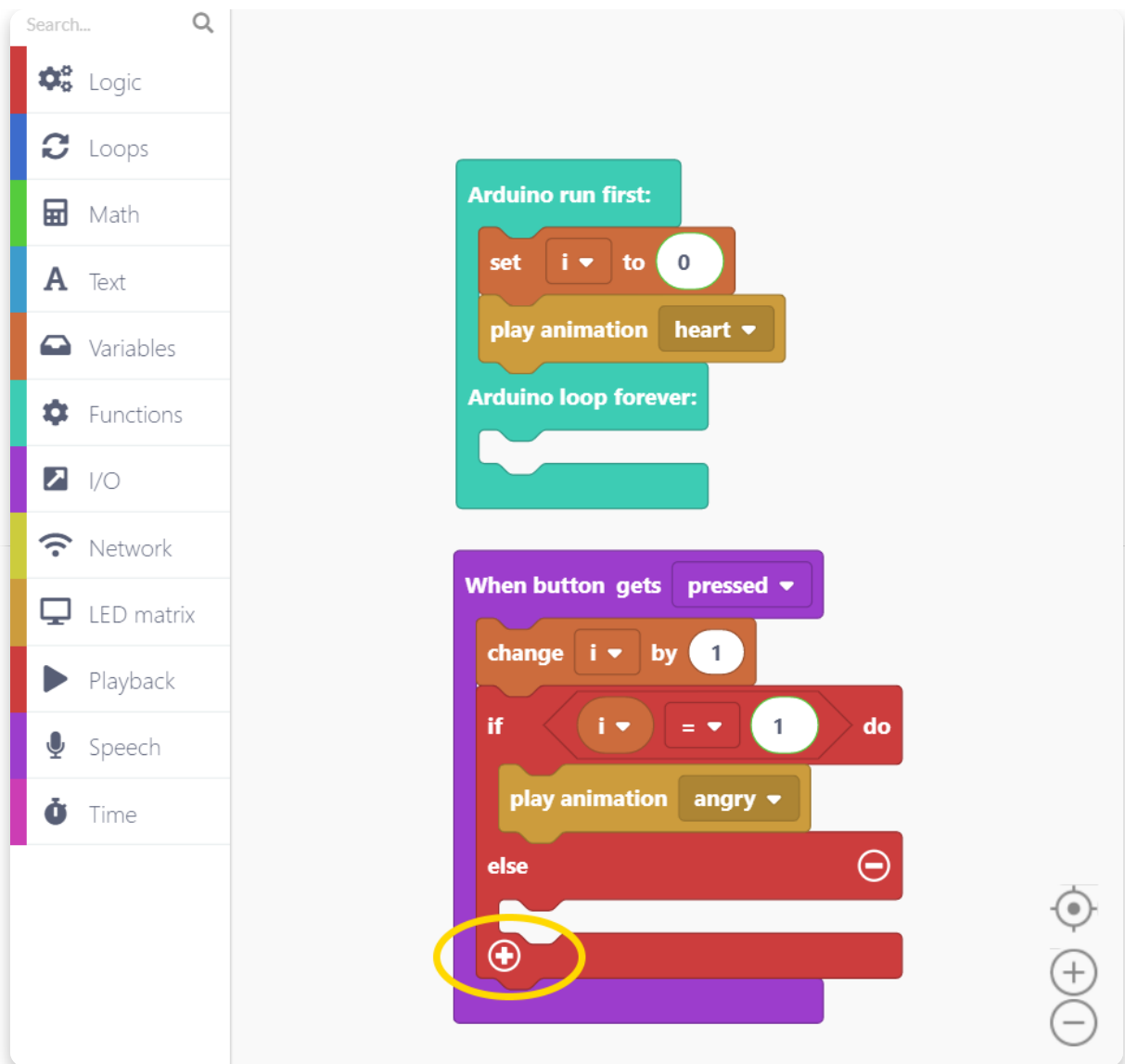
Tas, ko mēs darīsim tagad, ir likt Spenser rādīt dusmīgas sejas animāciju, ja mainīgais ir vienāds ar 1.

Jums jāievieto šāds bloks "Play animation" un nolaižamajā izvēlnē jāizvēlas "angry" animācija:

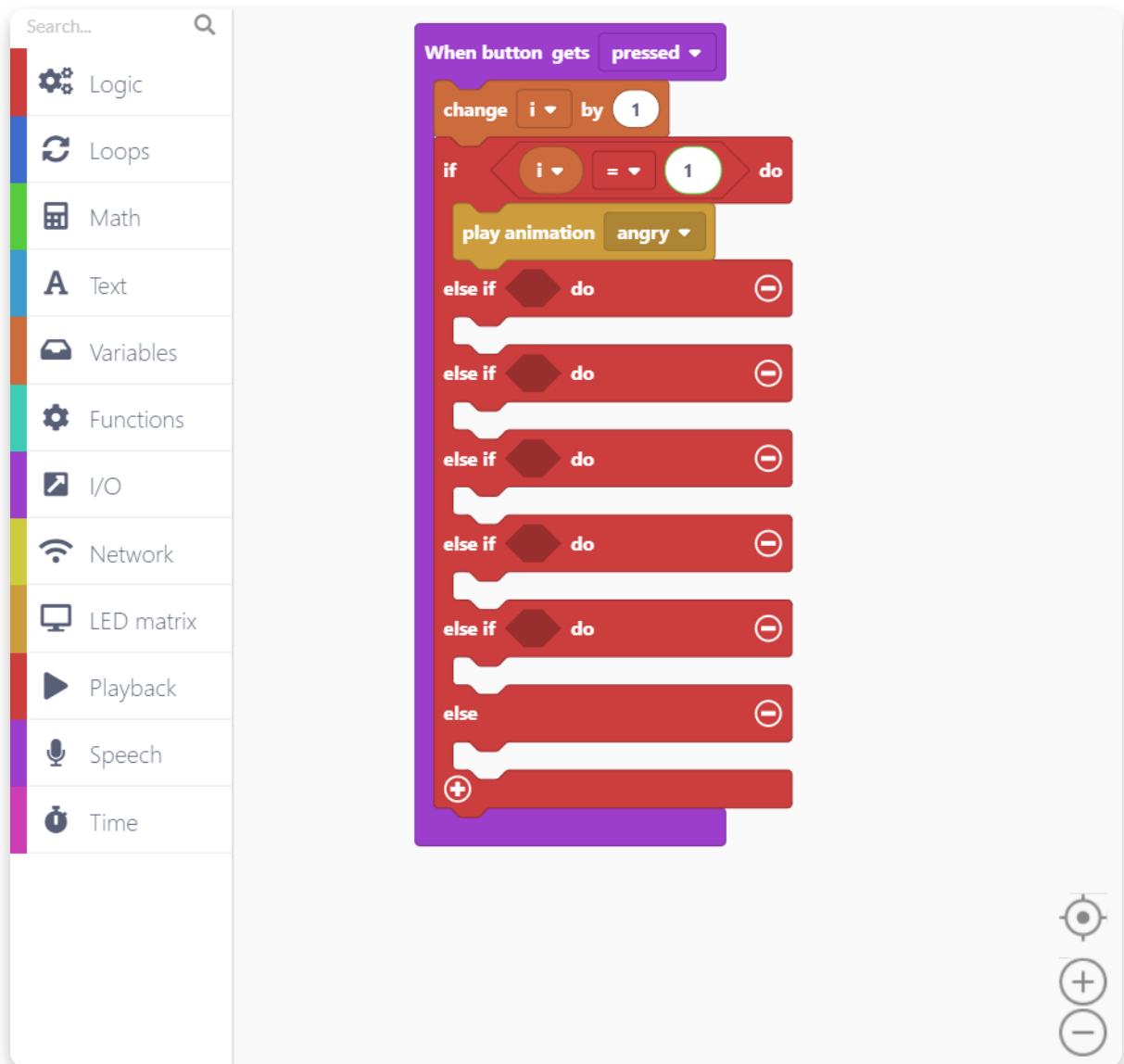


Liksim Spenser pārvietoties pa septiņām dažādām animācijām, nospiežot viņa lielo sarkano pogu.

Lai to izdarītu, jums piecas reizes jānoklikšķina uz baltās pluszīmes pogas loģiskā IF-THEN-ELSE bloka apakšējā kreisajā stūrī:

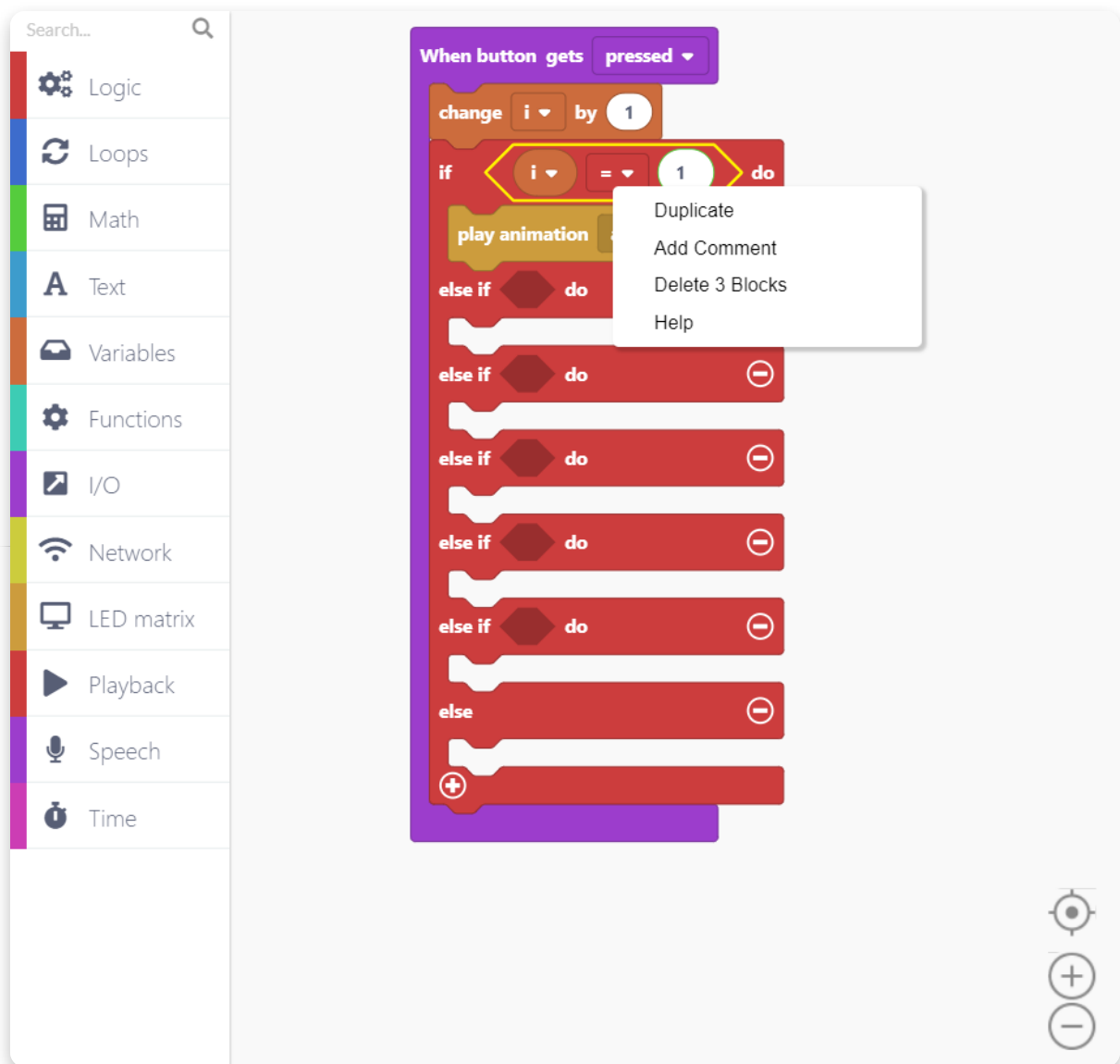


Tagad tam būtu jāizskatās šādi:



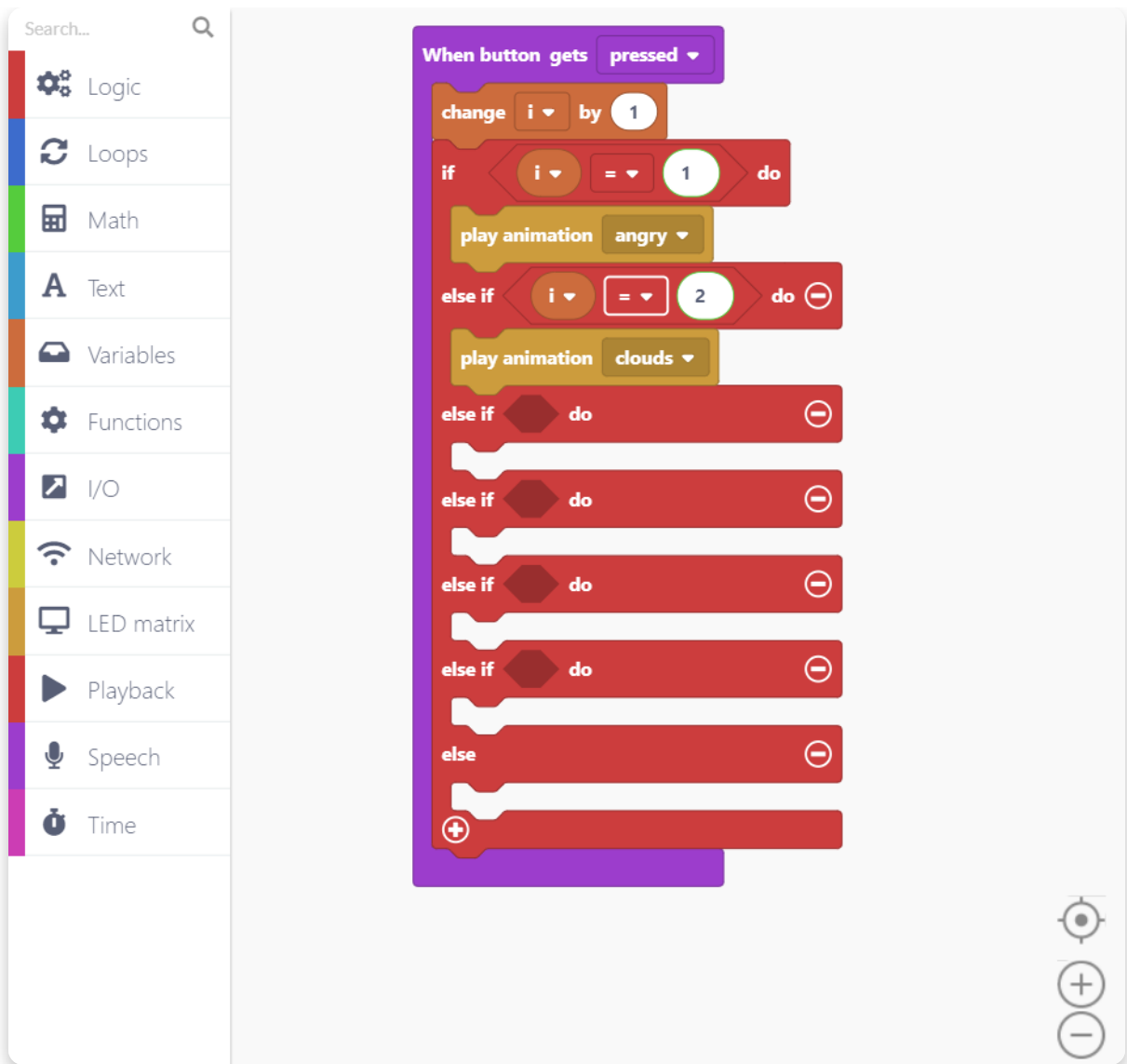
Ievietosim salīdzināšanas bloku katrā jaunizveidotajā loģikas nozarē.

Lai to izdarītu, varat vienkārši ar peles labo pogu noklikšķināt uz jau esošā salīdzināšanas bloka un noklikšķināt uz "duplicate".

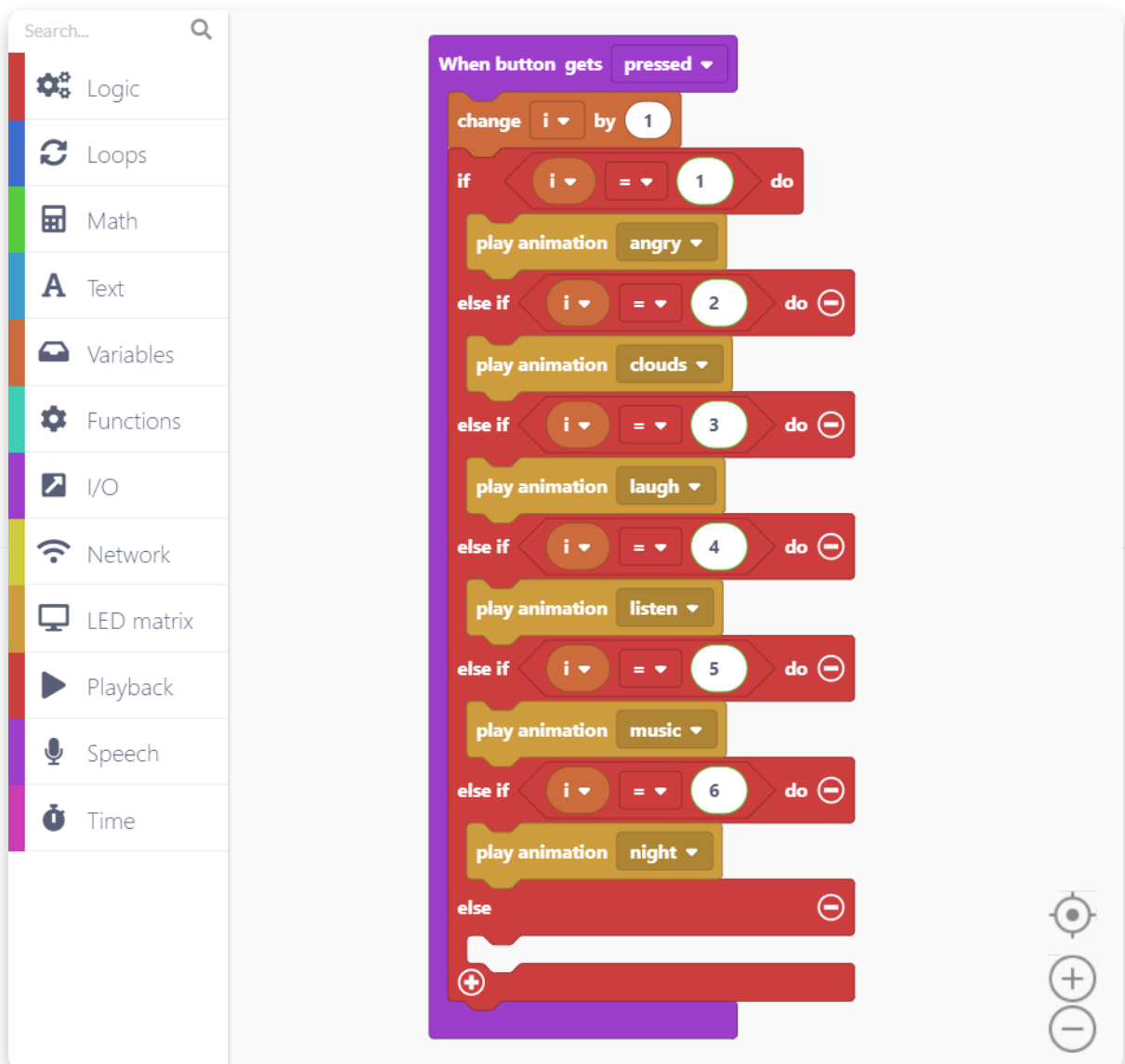


Varat arī nokopēt Play animācijas bloku un ievietot to šādi.

Liksīm Spenseram rādīt animāciju ar nosaukumu "clouds", kad mainīgais ir vienāds ar 2.

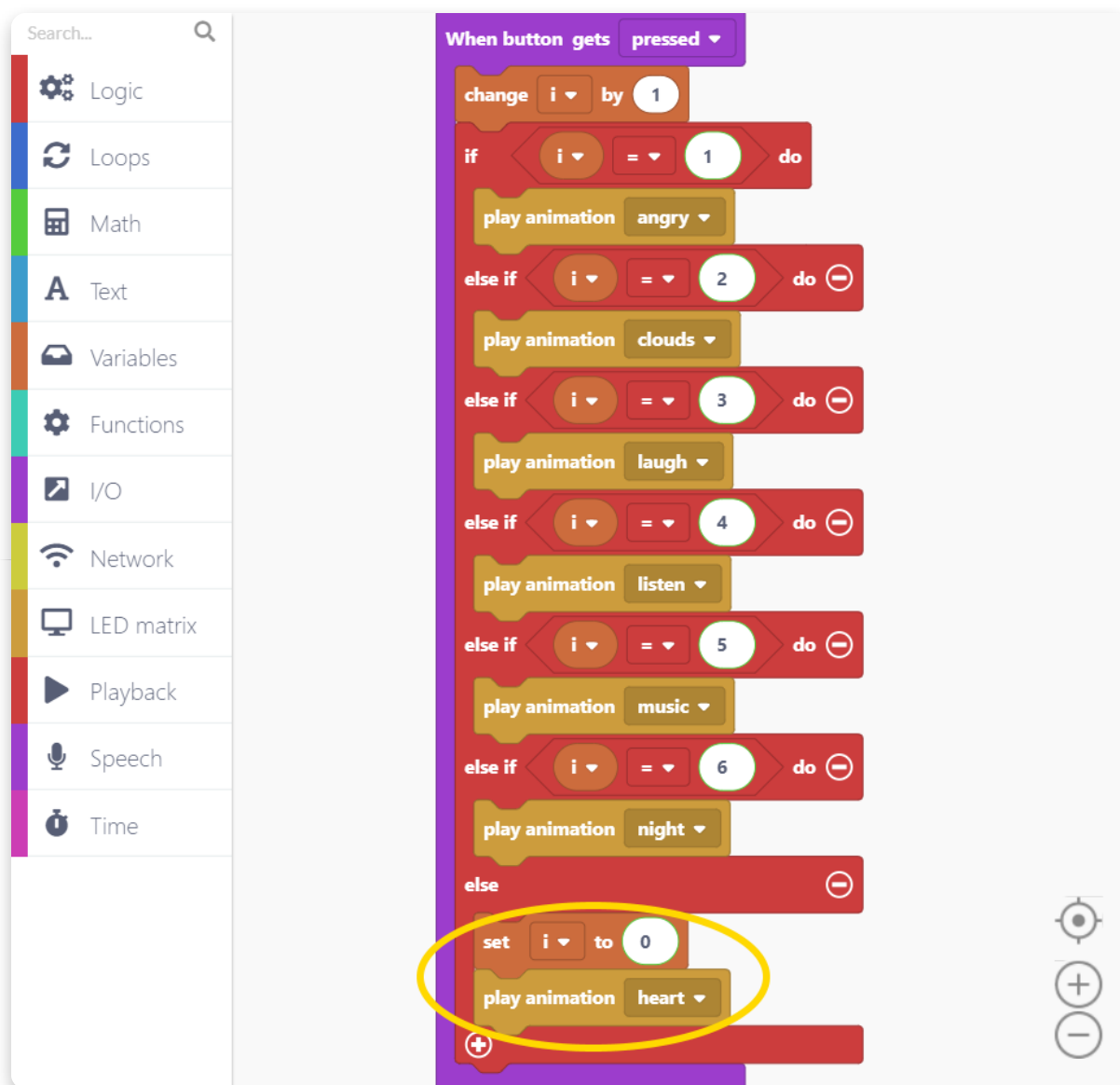


Es aizpildīšu arī pārējās šādas loģiskās filiāles un piešķiršu dažādām animācijām dažādām mainīgā vērtības:



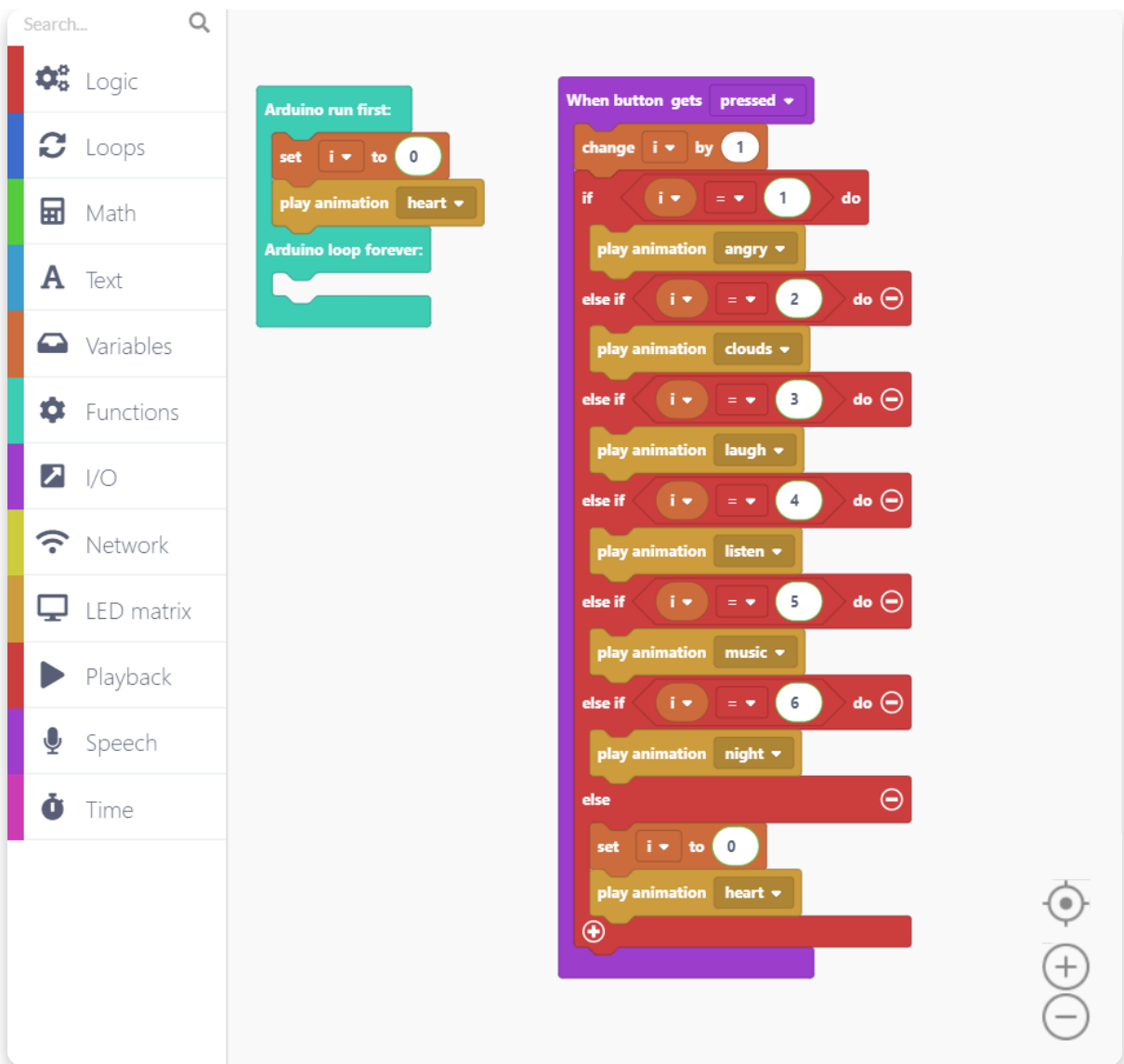
Ņem vērā – ja mēs vēlamies, lai Spenser apritētu šīs septiņas animācijas, mums ir jāatjauno mainīgā vērtība uz nulli, kad tā pārsniedz 6.

Mēs to varam izdarīt, vienkārši pievienojot šos divus blokus loģiskā bloka ELSE atzaram:

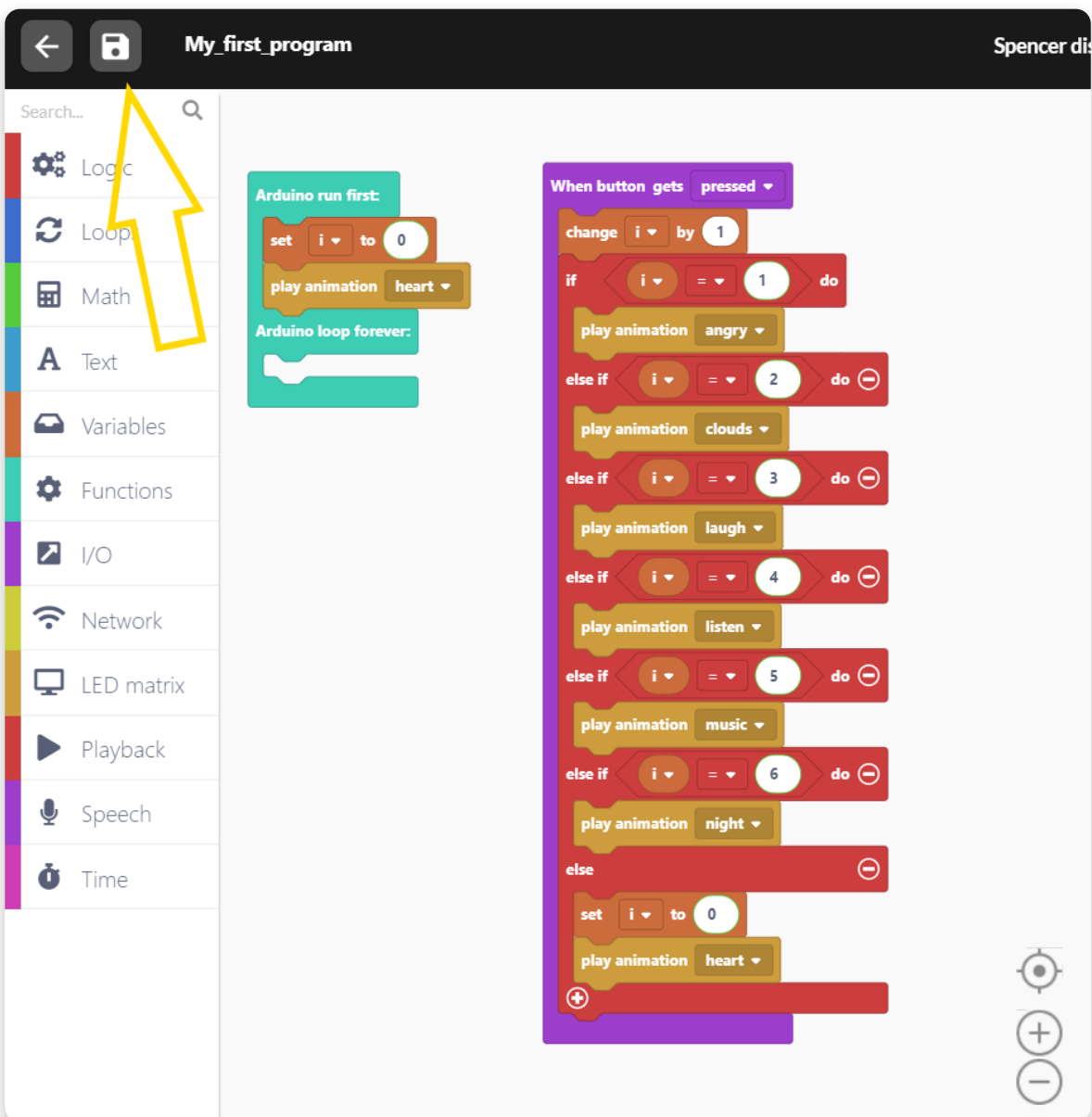


Šie divi bloki atiestatīs mainīgā vērtību uz nulli un atskaņos sākotnējo animāciju, ar kuru programma sākās ("heart" animācija).

Ja esat visu pareizi ievērojis, koda beigās vajadzētu izskatīties šādi:



Neizmirstiet ietaupīt visu savu smago darbu!



Tagad nospiediet lielo sarkano pogu "Palaist" un pārbaudiet kodu pats.

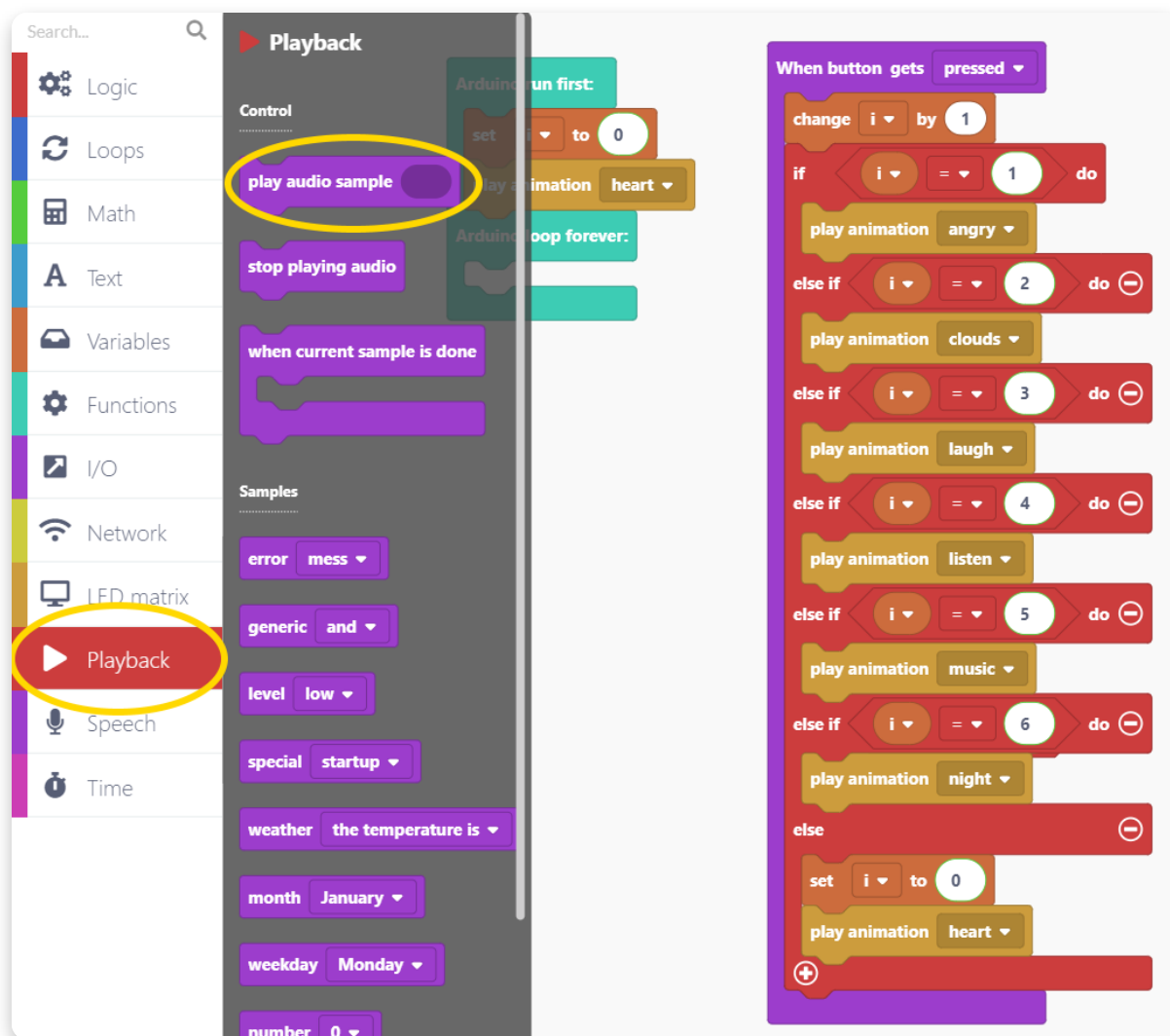
Nu ir laiks apskatīt ko vel mēs varam izdarīt ar Spencer.

Ahoy skaņas efekti.

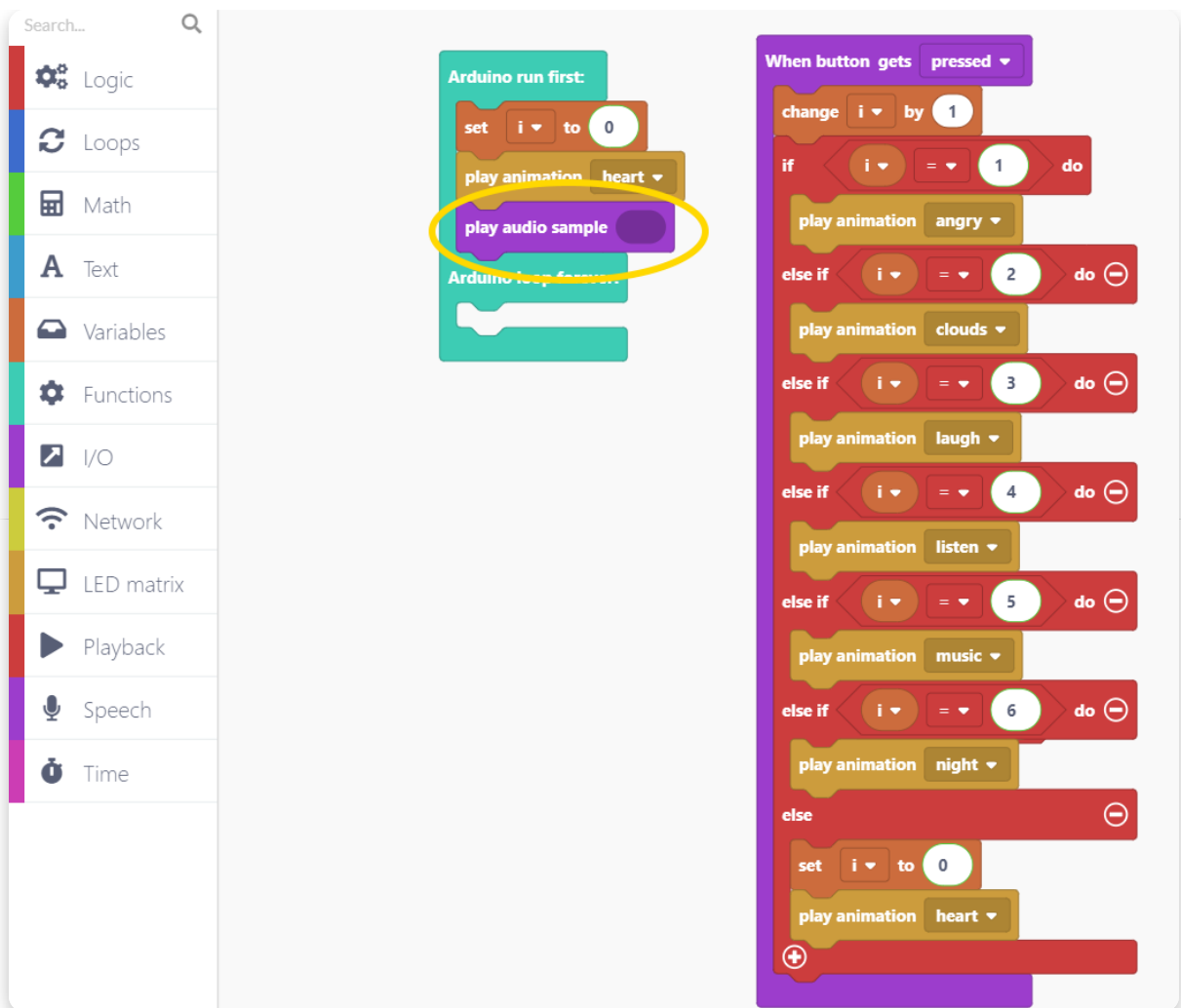
Spencers var arī sarunāties un radīt smieklīgus trokšņus. Tam ir iepriekš ielādēti dažādi skaņu efekti un dialogi.

Jūs varat augšupielādēt arī savus balss dialogus, bet vispirms atskaņosim dažus iepriekš ielādētos skaņas paraugus.

Sadaļā ar nosaukumu "Playback" ir viss, kas saistīts ar skaņu. Sāksim ar bloka "Play audio sample" vilkšanu un nomešanu zīmēšanas laukumā.



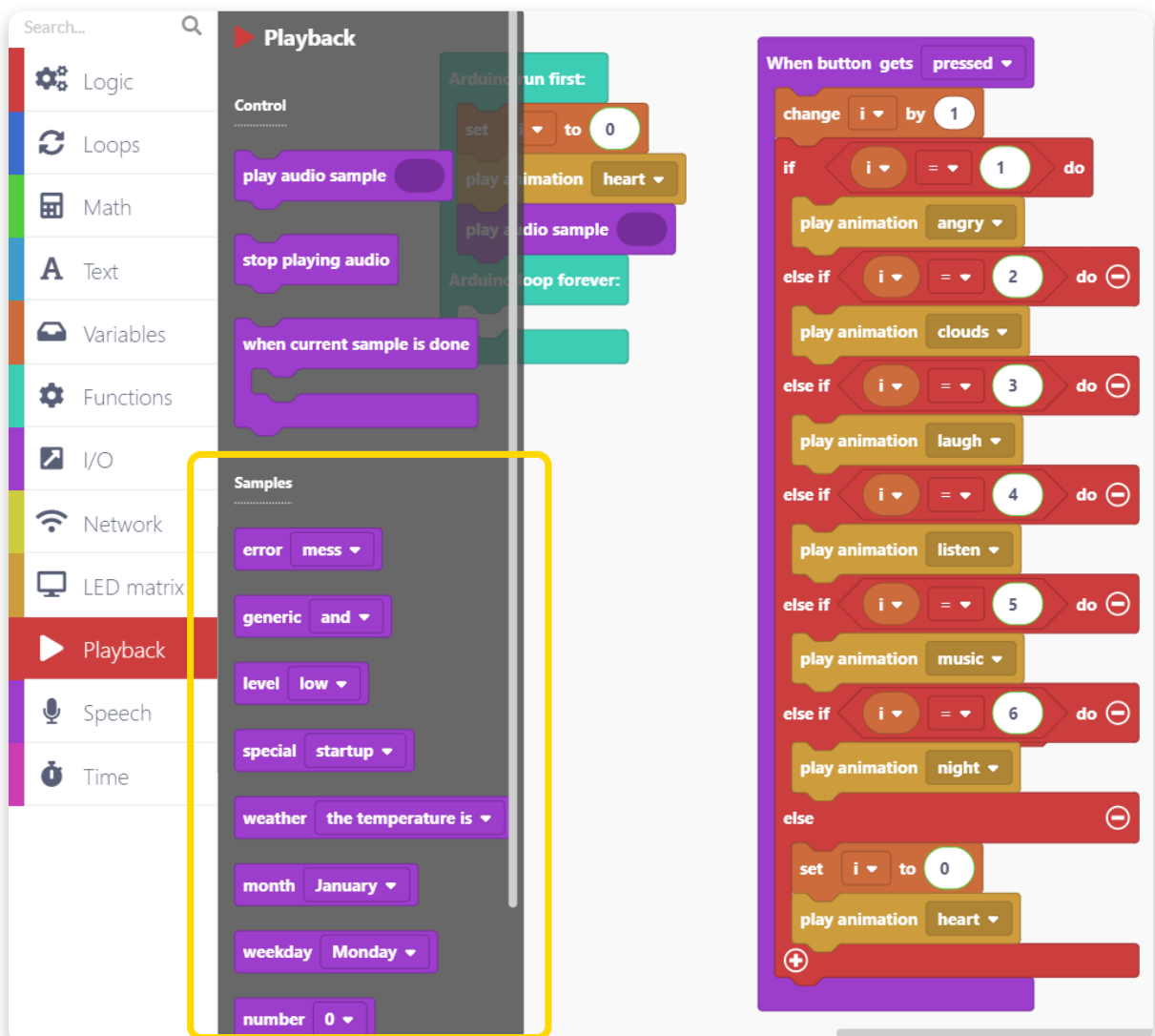
levietojiet bloku "Arduino run first" filiālē tā, lai skaņas efekts tiktu atskaņots, tiklīdz sākas jūsu programma.



Tālāk mums jāizvēlas skaņas efekts, kas tiks atskaņots.

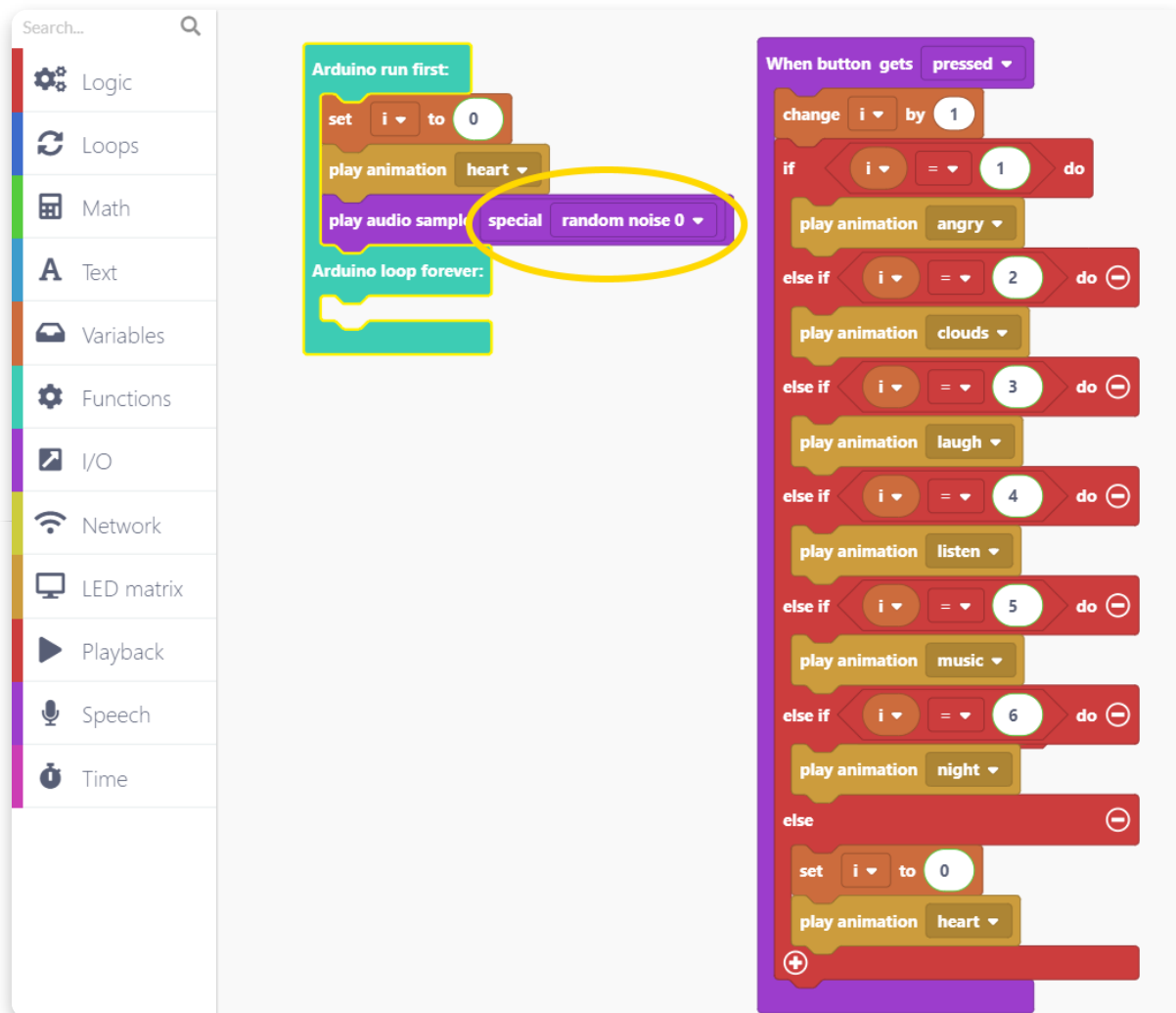
Ir vesela sadaļa dažādu paraugu, kurus varat izvēlēties un izpētīt.

Es ņemšu parauga bloku ar nosaukumu "Special".



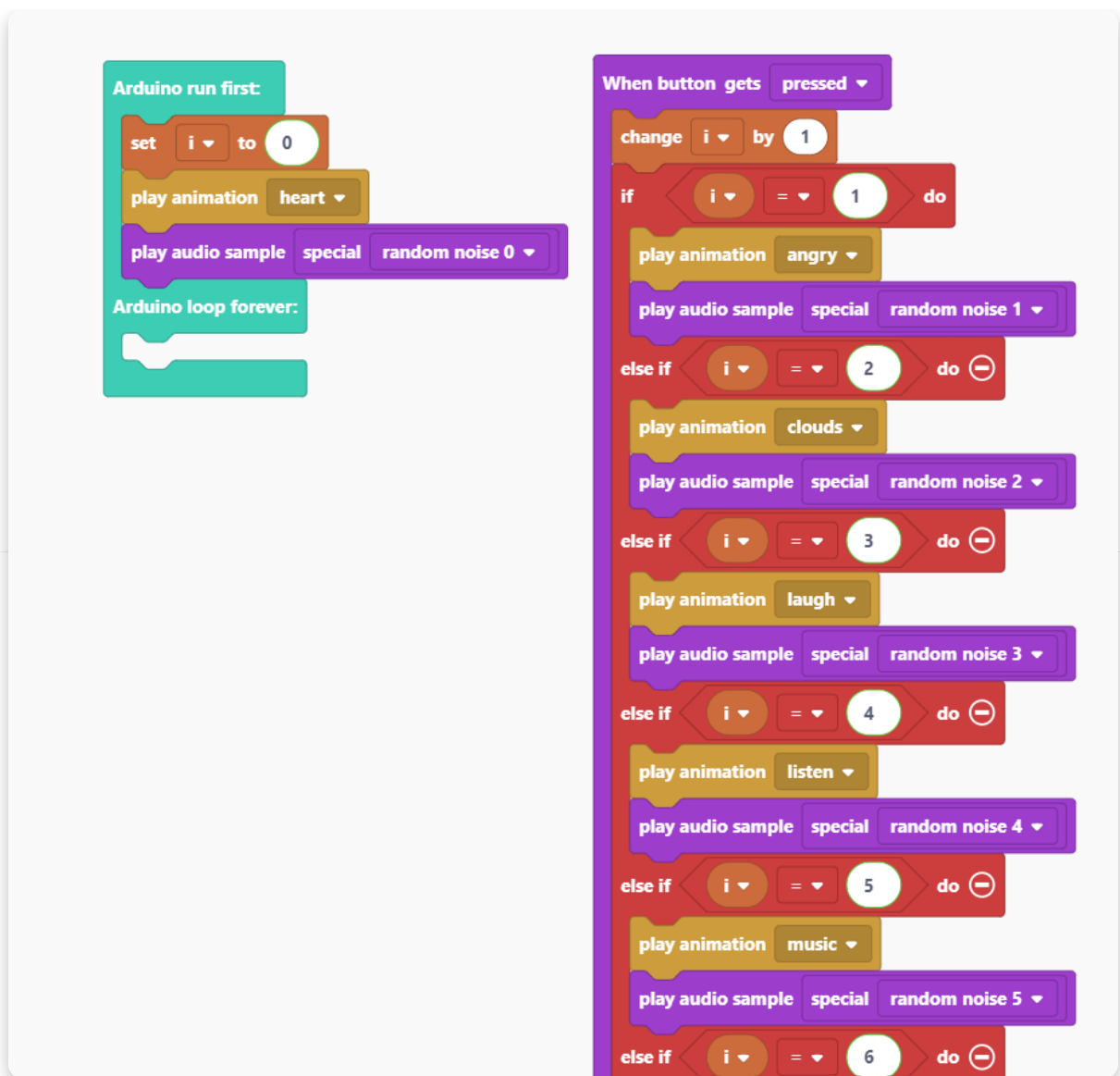
Šis bloka paraugs jāievieto blokā "Play audio sample". Nospiežot nolaižamo izvēlni, varat izvēlēties dažādus audio paraugus, kas atrodas audio paraugu

kategorijā "Special".



Ja palaidīsiet šo kodu un augšupielādēsiet to savā Spencer, tas atskaņos izvēlēto skaņas efektu, tiklīdz sāksies jūsu programma.

Ja mēs vēlamies pievienot nedaudz vairāk mūsu kodam, mēs varam likt Spenseram atskaņot citu audio paraugu katru reizi, kad nospiežat tā pogu. Jūsu kodam vajadzētu izskatīties šādi, ja vēlaties to darīt:



Atcerieties saglabāt kodu!

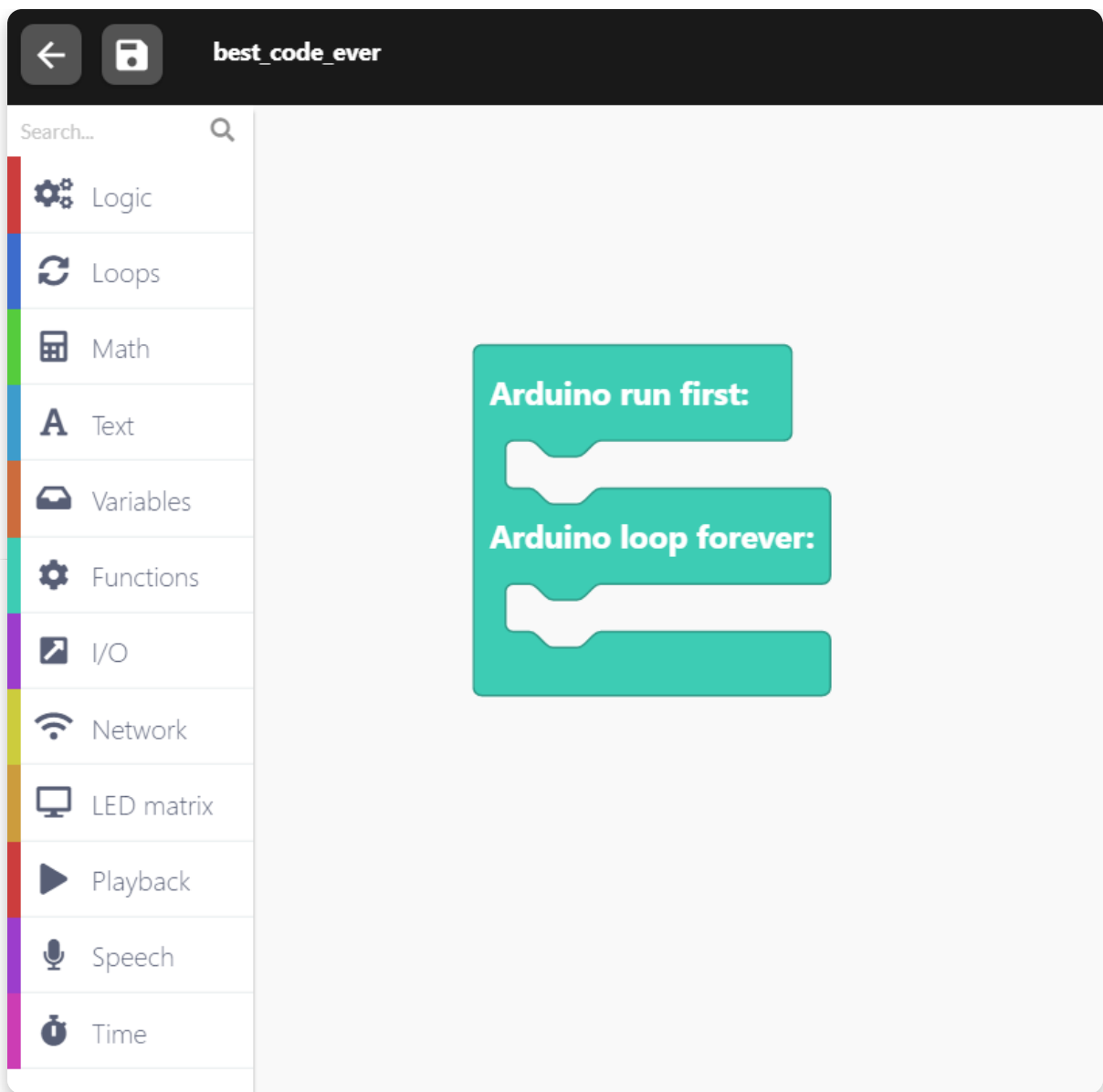
Tas bija jautri. Apskatīsim, ko vēl mēs varam darīt.

[Vairāk kodēšanas](#)

Laiks likt Spencer runāt

Tagad izmēģināsim kaut ko modernāku - mēs liksim Spenseram ģenerēt un "pateikt" unikālu dialoga katru reizi, kad uzsitīsim viņa lielo sarkano pogu.

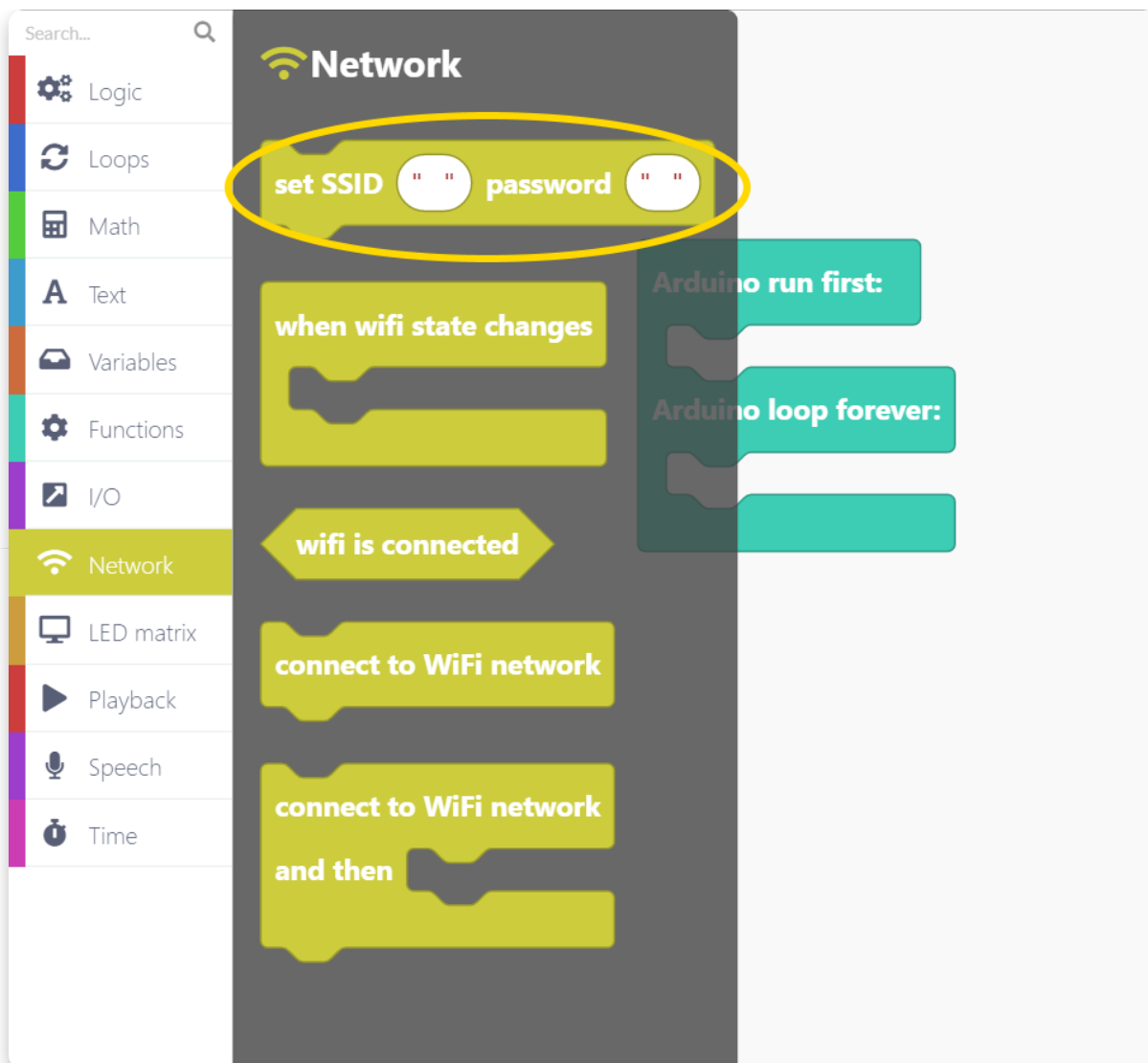
Sāciet no jauna, izveidojot jaunu projektu. Saglabājiet to uzreiz un nosauciet to par kaut ko interesantu (es savu esmu nosaucis par "best_code_ever").



Tālāk mums jāsavieno Spenser ar WiFi tīklu.

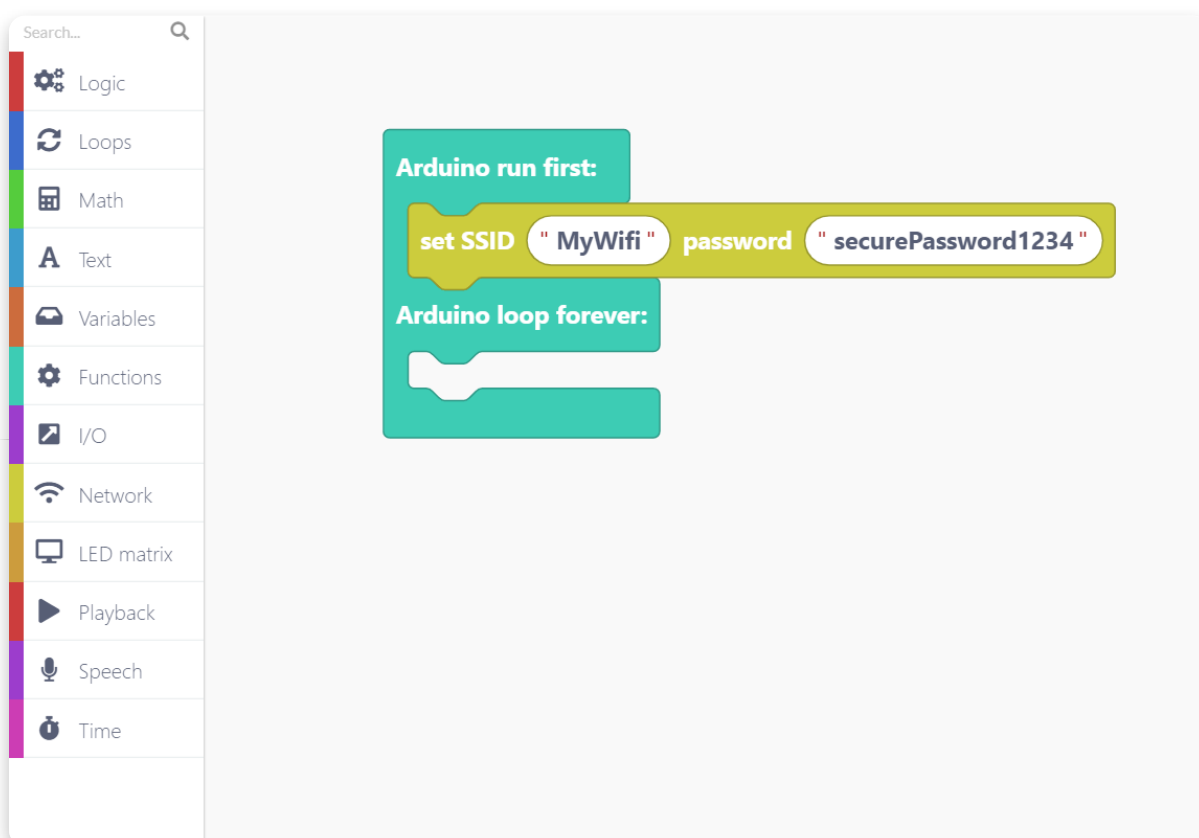
Spenser ir nepieciešama piekļuve internetam, lai "runātu". Tas darbojas tā, ka Spenser izveido savienojumu ar mūsu serveri un nosūta tekstu, ko vēlas pateikt. Pēc tam serveris ģenerē audio klipu, izmantojot TTS algoritmu, un nosūta to atpakaļ uz jūsu Spenser.

Lai izveidotu savienojumu ar WiFi tīklu, jums būs nepieciešams bloks "Set SSID and password", kas atrodas kategorijā "Network".

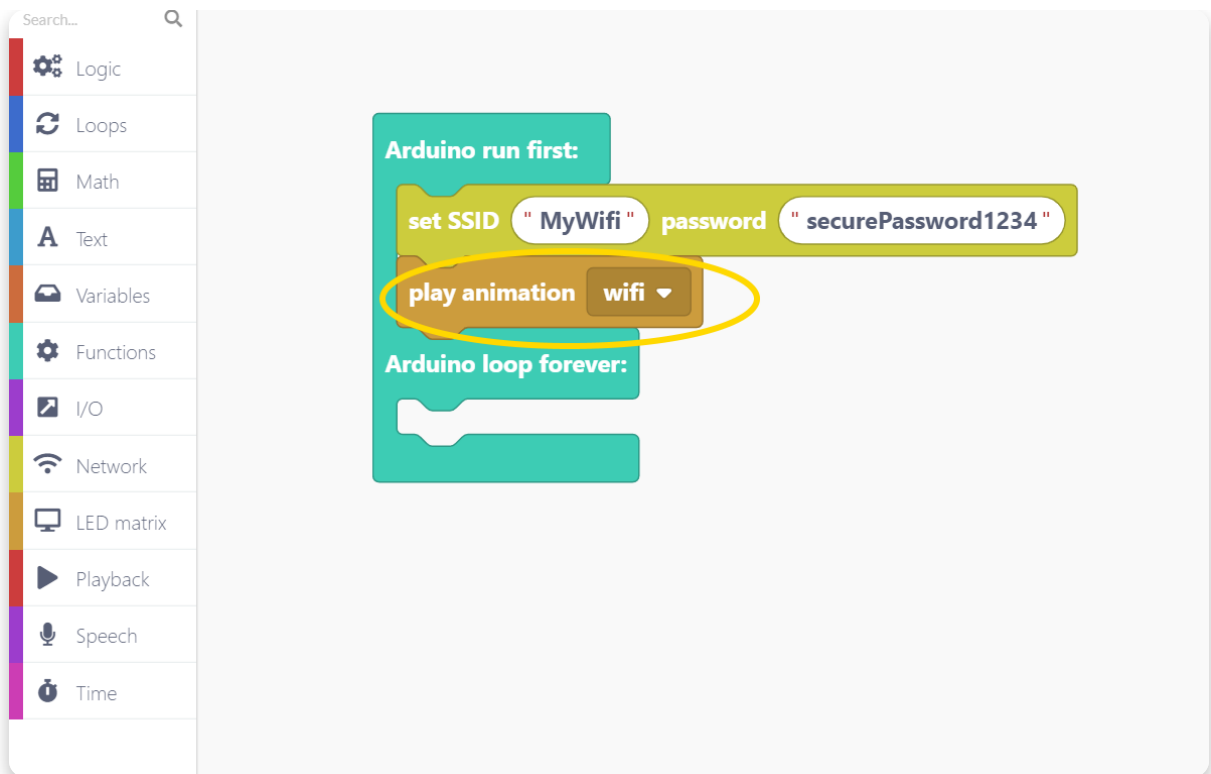


Velciet bloku zīmēšanas apgabalā un ievadiet savu WiFi tīkla nosaukumu un paroli.

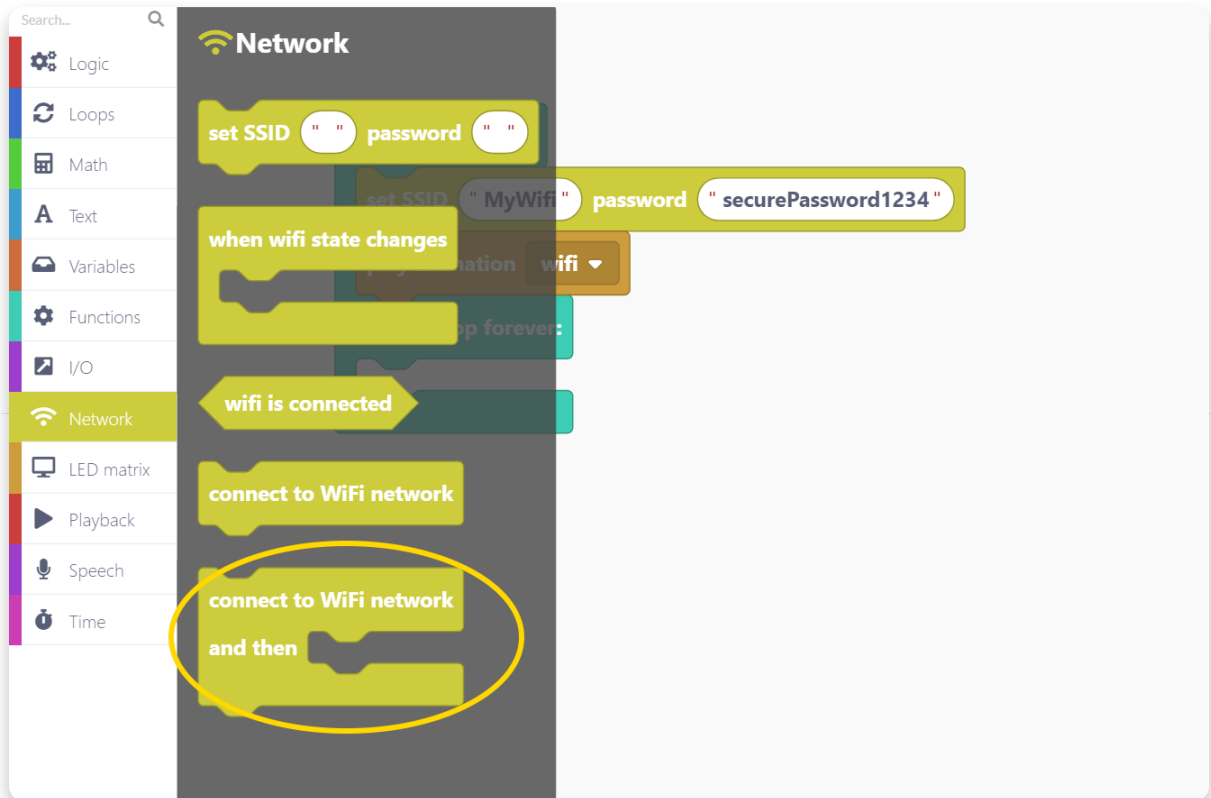
SSID ir jūsu tīkla nosaukums.



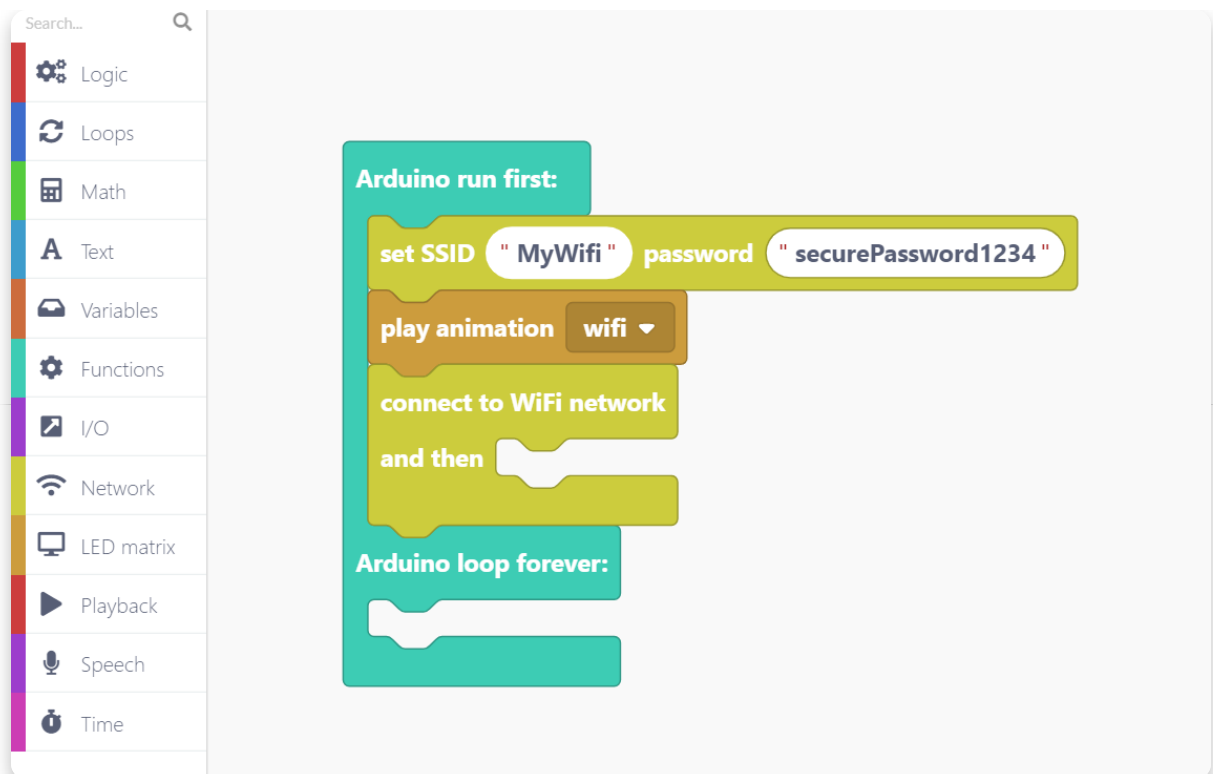
Kad mēs iestatīsim tīkla pieteikšanās datus, liksim Spenser parādīt animāciju "WiFi". Mēs varam to izdarīt, pievienojot šo bloku šeit:



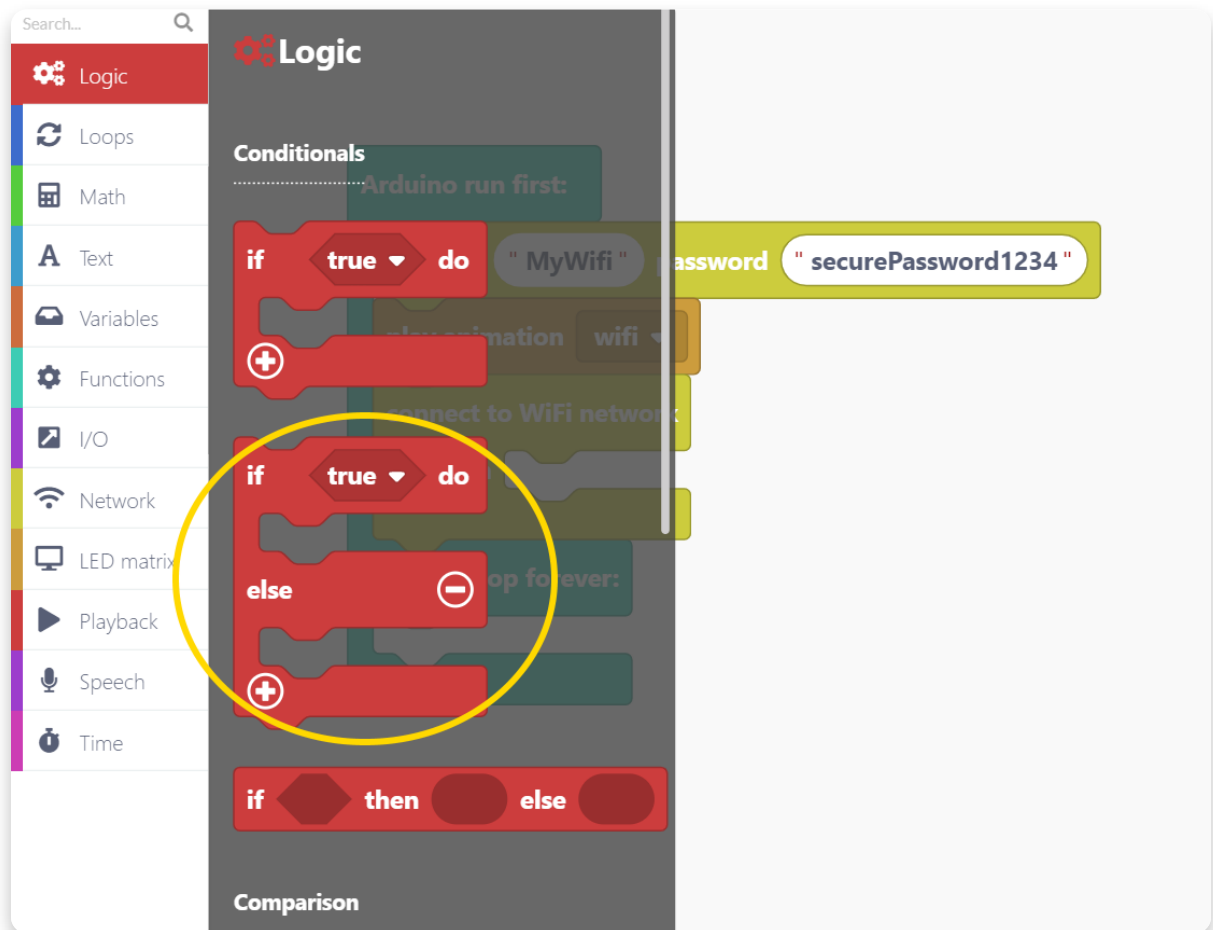
Tagad jums ir jāvelk un nometiet bloku "Connect to WiFi network". Šis bloks sāk savienojumu.



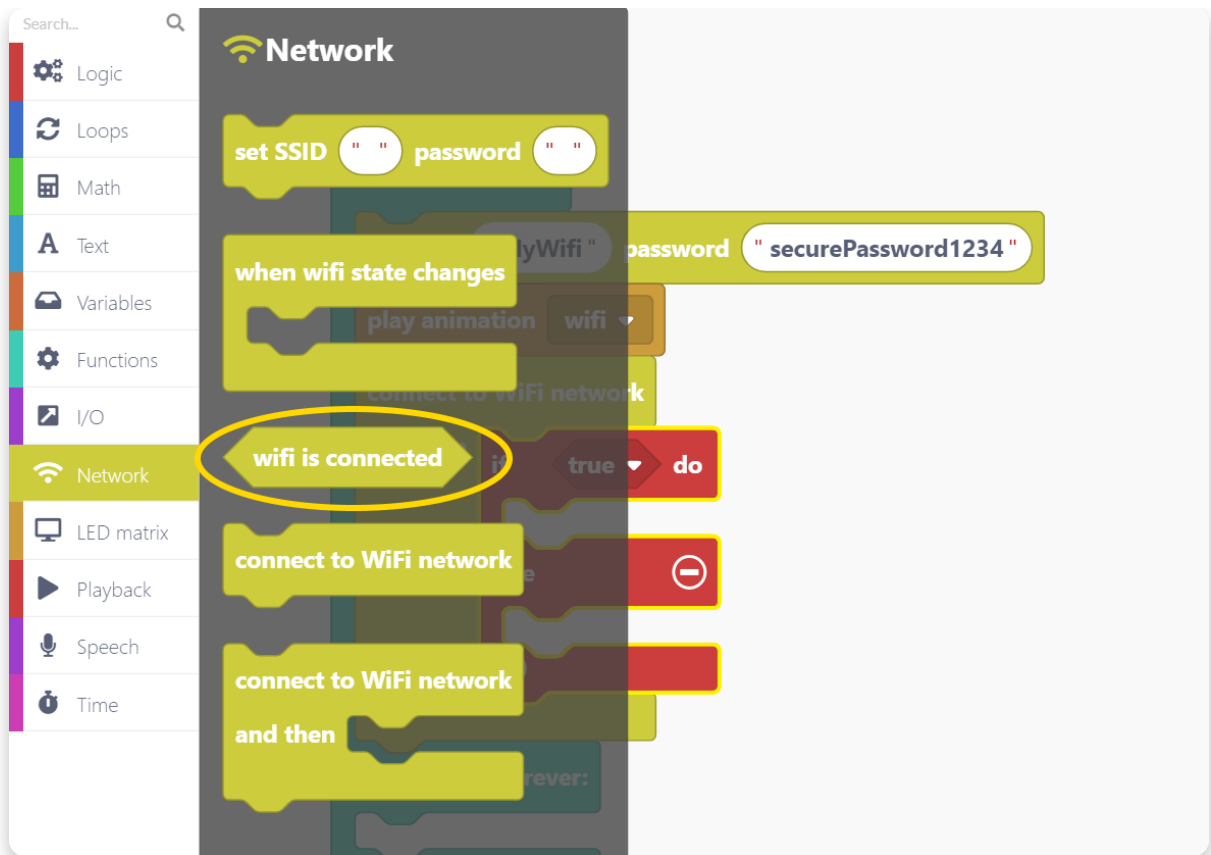
Novieto bloku šeit:



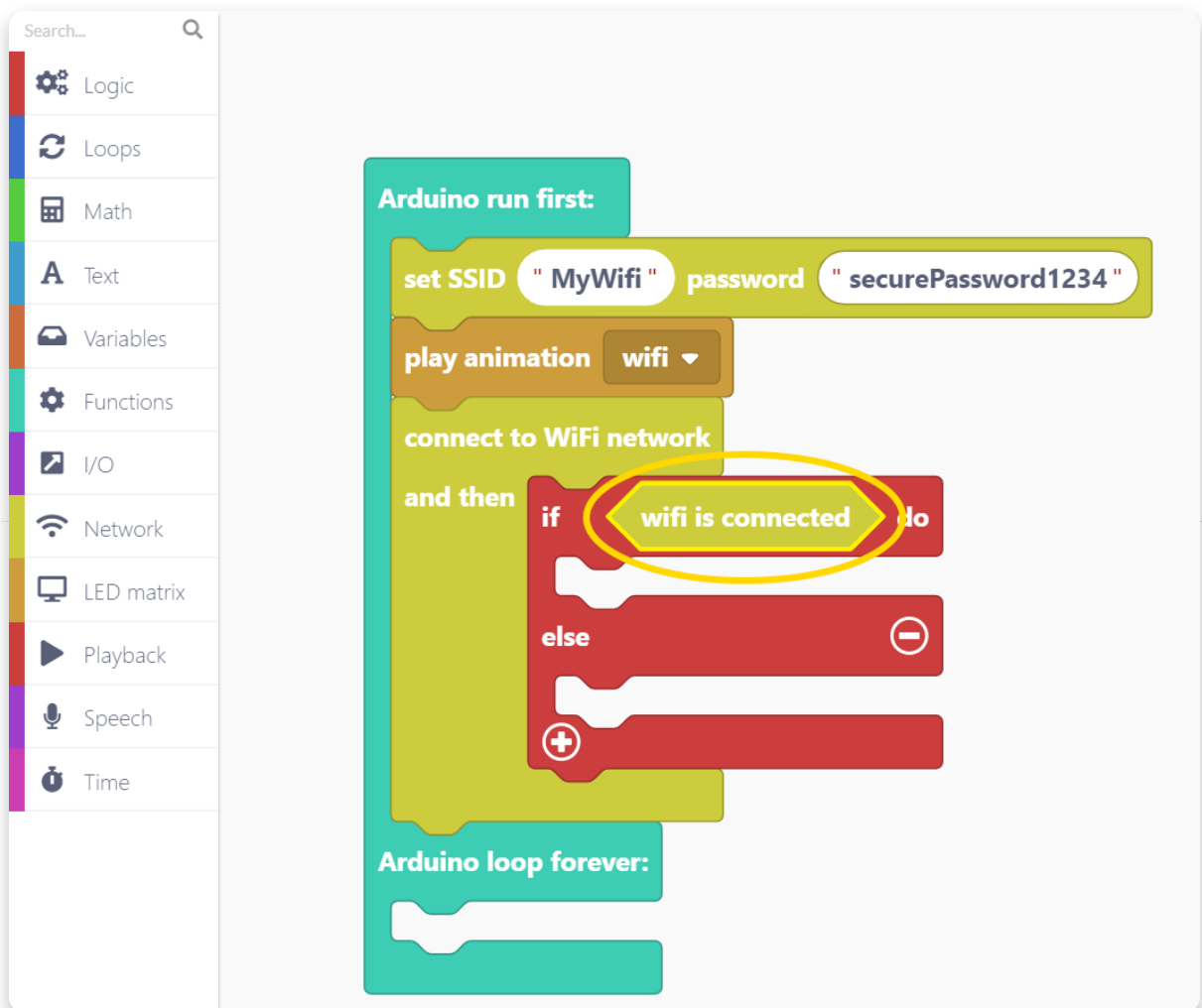
Tagad mums būs nepieciešams loģiskais bloks IF-THEN-ELSE. Mēs izmantosim šo bloku, lai pārbaudītu, vai WiFi tīkla savienojums ir veiksmīgi izveidots.



Sadaļā "network" izmantojiet nosacījumu "wifi is conected".

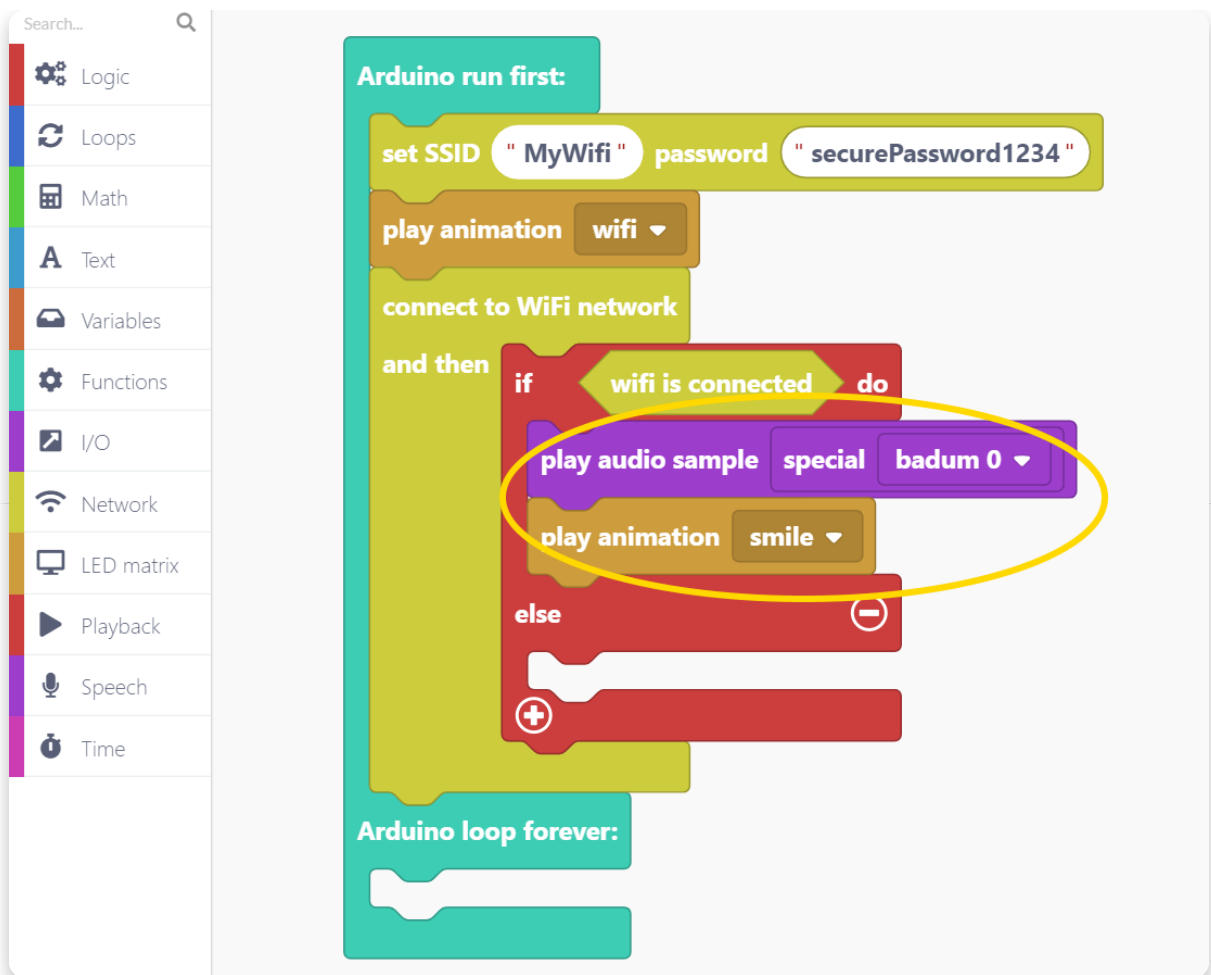


Novietojiet to loģiskajā blokā IF-THEN-ELSE kā šādu nosacījumu:

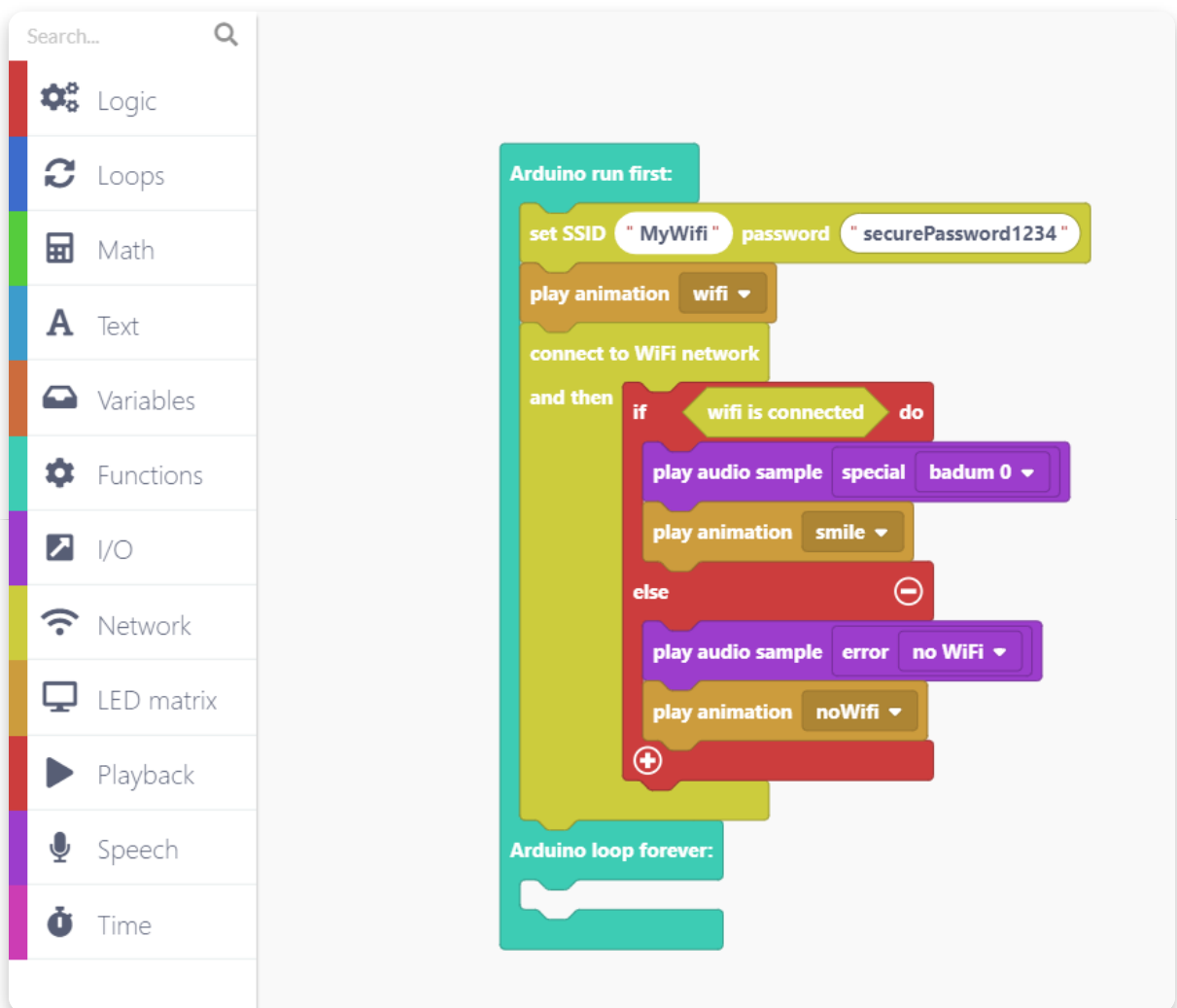


Spēlēsīm noteiktu skaņas paraugu un animāciju, ja wifi savienojums ir veiksmīgi izveidots.

Esmu izvēlējis smaidošo animāciju un skaņas efektu ar nosaukumu "badum 0", jo tas izklausījās smieklīgi:



Ja WiFi interneta savienojums nav veiksmīgi izveidots, atskaņosim animācijas un audio paraugu ar nosaukumu "no WiFi".

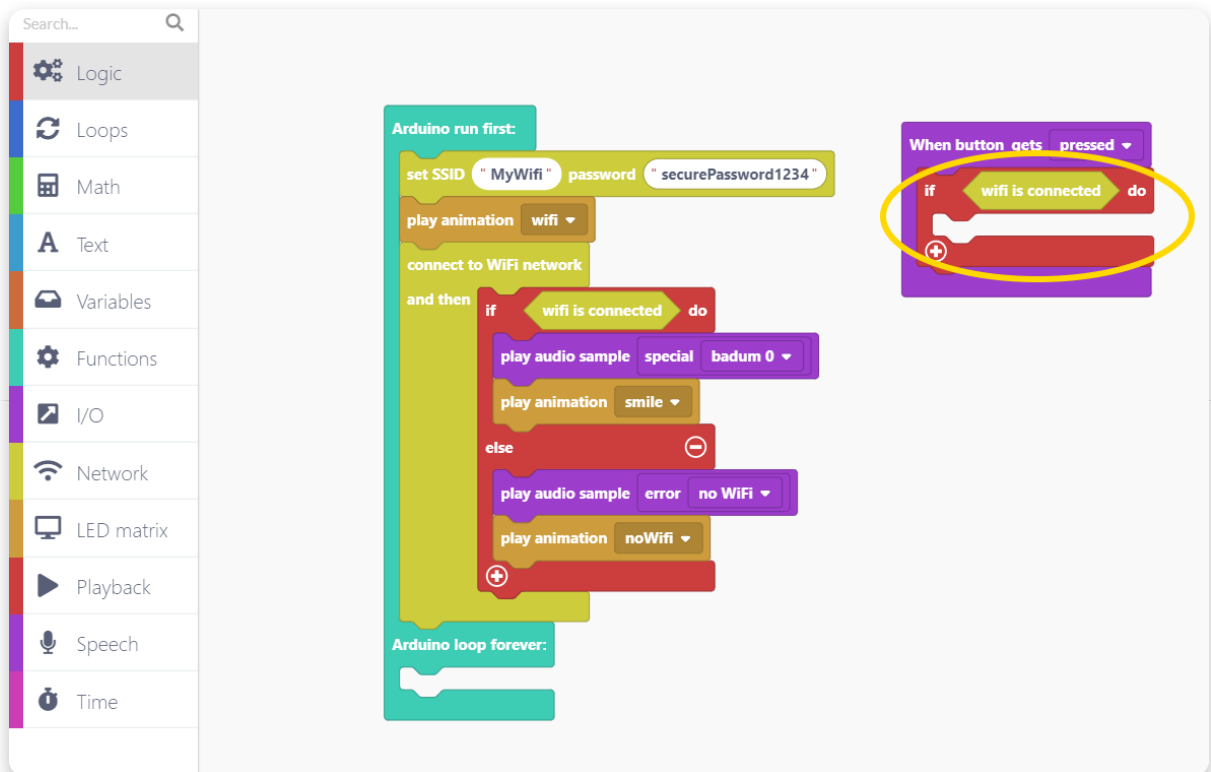


Šī programma darbosies, ja Spensers veiksmīgi izveido savienojumu ar tīkliem. Ja nē, Spensers pateiks, ka tai ir problēmas ar savienojumu ar WiFi tīkliem.

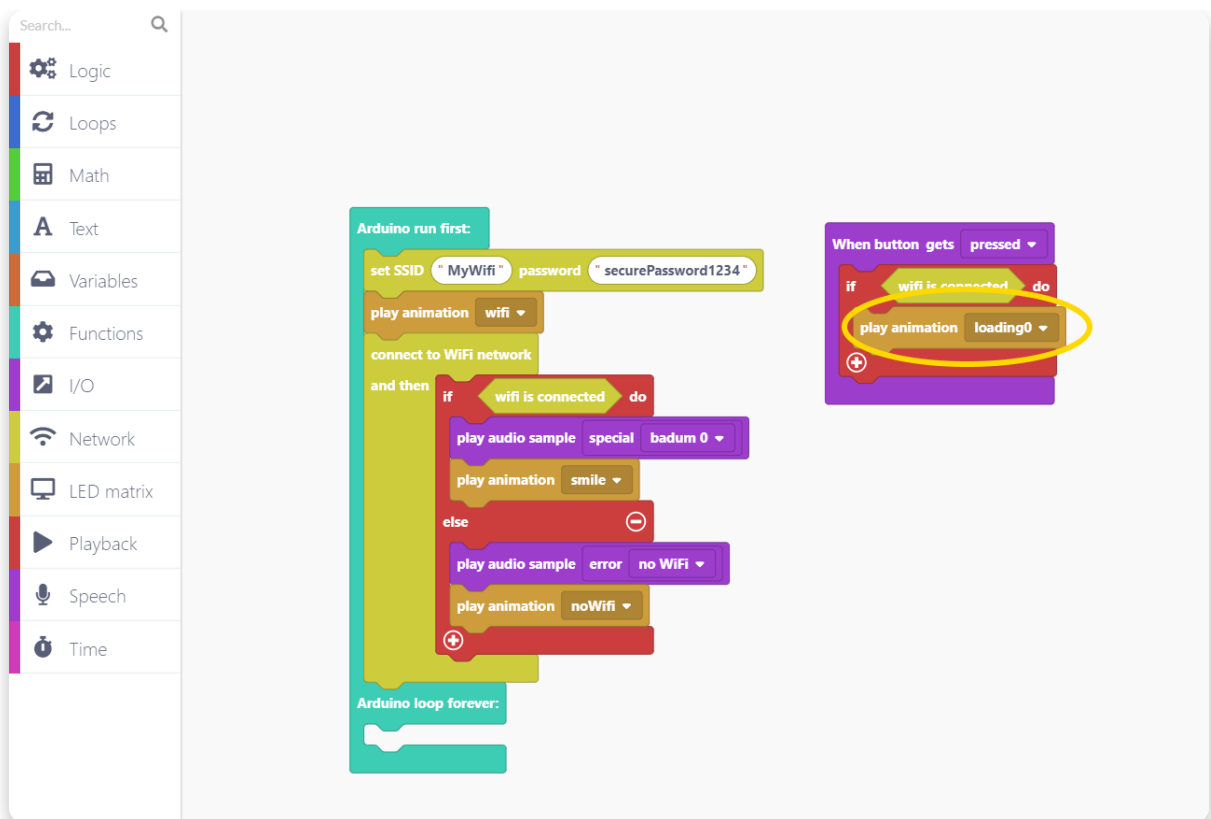
Tagad pievienosim notikumu bloku "When button gets pressed":



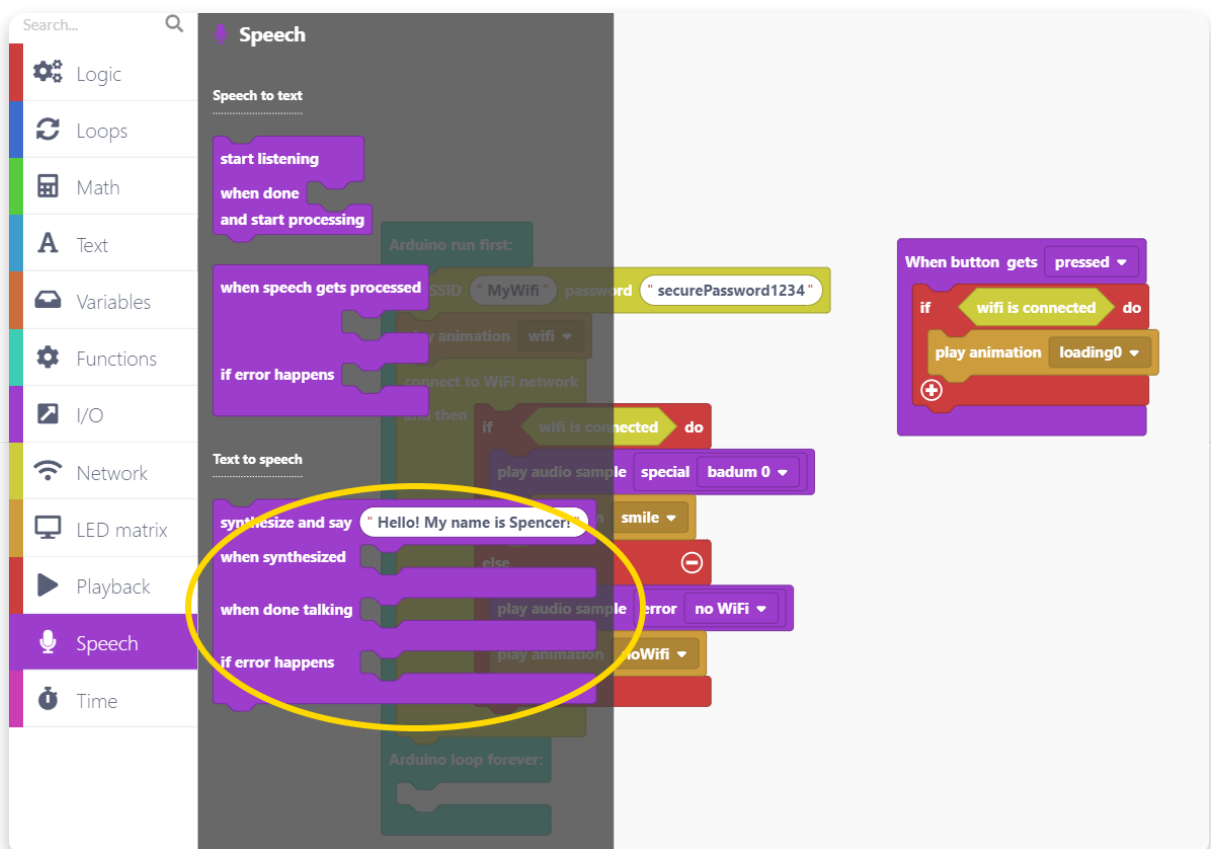
Katru reizi, kad tiek nospiesta poga, pārbaudiet, vai WiFi ir izveidots savienojums. Dariet to ar to pašu loģisko bloku IF-THEN un kā nosacījumu ievietojiet "wifi is connected":



Spenseram būs vajadzīgs īss laiks, lai izveidotu vēlamo audio paraugu. Spēlēsīm ielādes animāciju, kamēr paraugu ģenerē:



Lai veiktu nākamo darbību, atveriet kategoriju ar nosaukumu "Speech" un atrodi bloku "synthesize and say".



Novietojiet bloku šeit un ierakstiet teikumu, kuru vēlaties pateikt Spenseram. Esmu ierakstījis "Hellow world. This is my first program".



Kad dialoga balss paraugs tiek ģenerēts (sintezēts), tas tiks automātiski atskaņots.

Kad tas notiks, padarīsim Spensera seju tādu, kā viņš runā, spēlējot animāciju ar nosaukumu "talk".

Kad Spensers pabeidz sarunu, es spēlēšu animāciju ar nosaukumu "wink".



Neaizmirstiet saglabāt savu smago darbu, nospiežot pogu Saglabāt augšējā kreisajā stūrī.

Palaidiet kodu un augšupielādējiet to savā Spenser. Katru reizi, nospiežot Spensera pogu, tam vajadzētu skaļi nolasīt tekstu, kuru esat ierakstījis blokā "Synthisize and say".

Ja jūsu Spenser ir problēmas ar savienojuma izveidi ar Wi-Fi tīklu, pārbaudiet tīkla nosaukumu un paroli un mēģiniet izslēgt un ieslēgt spencer.



Atceries, Spenser strādā tikai ar 2.4GHz (IEEE 802.11 b/g/h) WiFi signālu. 5GHz (IEEE 802.11 ac) tīkli, diemžēl, nav atbalstīti.

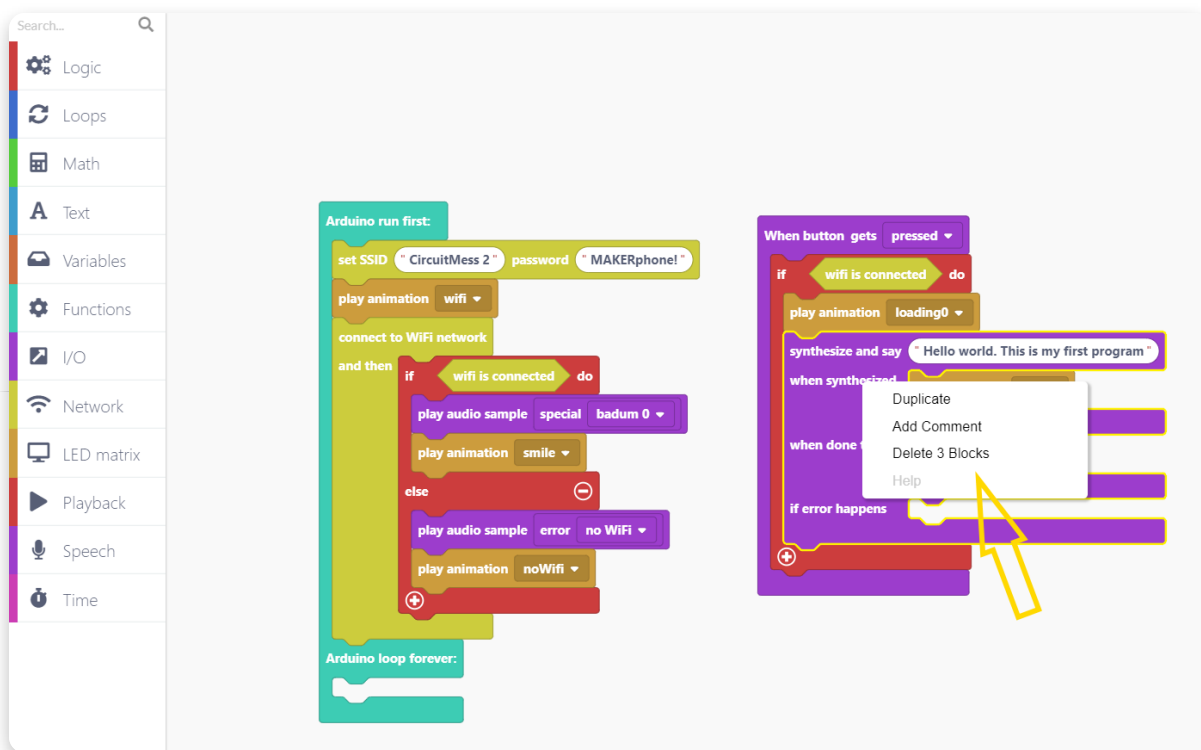
Ja procesa laikā gadās kādas problēmas, tad sazinaties ar mums contact@circuitmess.com.

Tagad liksim Spencer klausīties un atkārtot!

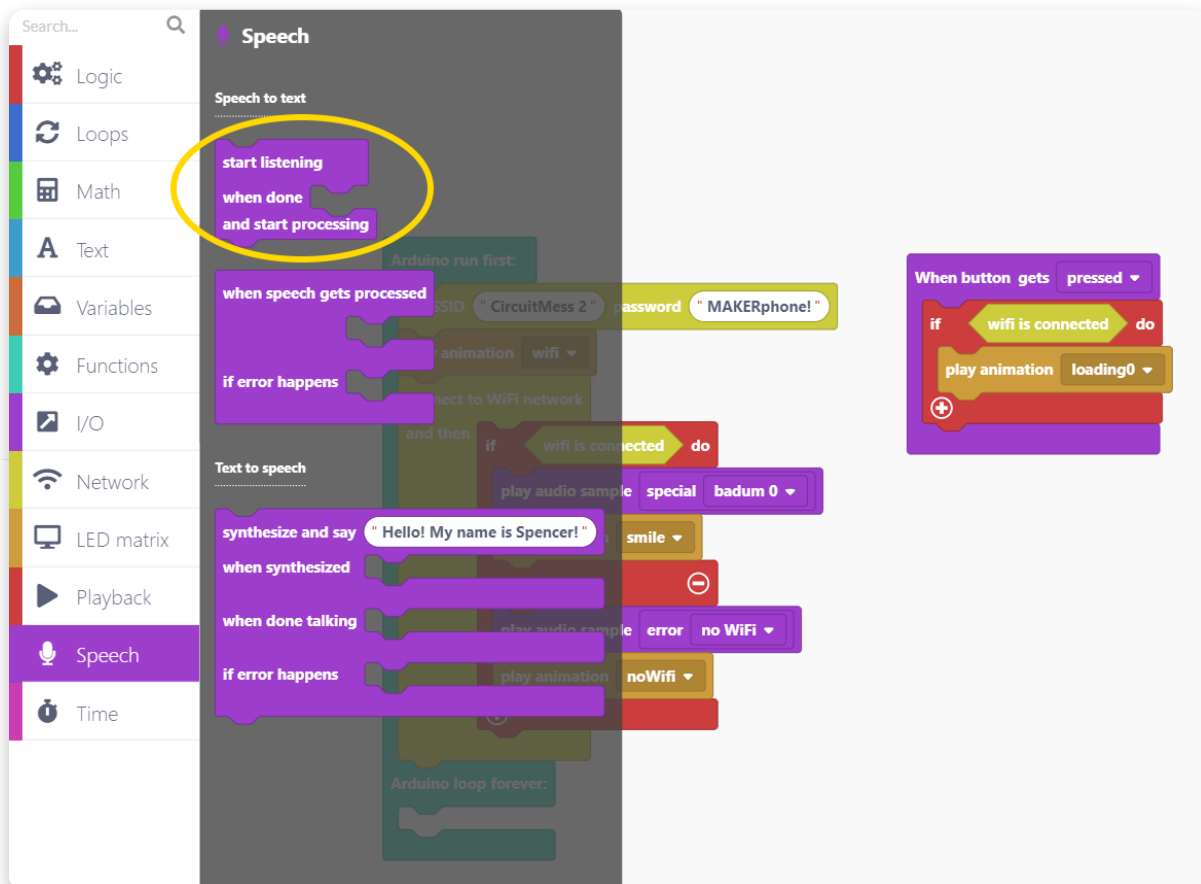
Lai iegūtu vislielāko kodēšanas izaicinājumu, liksim Spenseram klausīties mūsu teikto un atkārtot šo pašu teikumu ar savu smieklīgo robotu balsi.

Mēs varam vienkārši turpināt kodu, kuru esat izveidojīs iepriekšējā nodaļā.

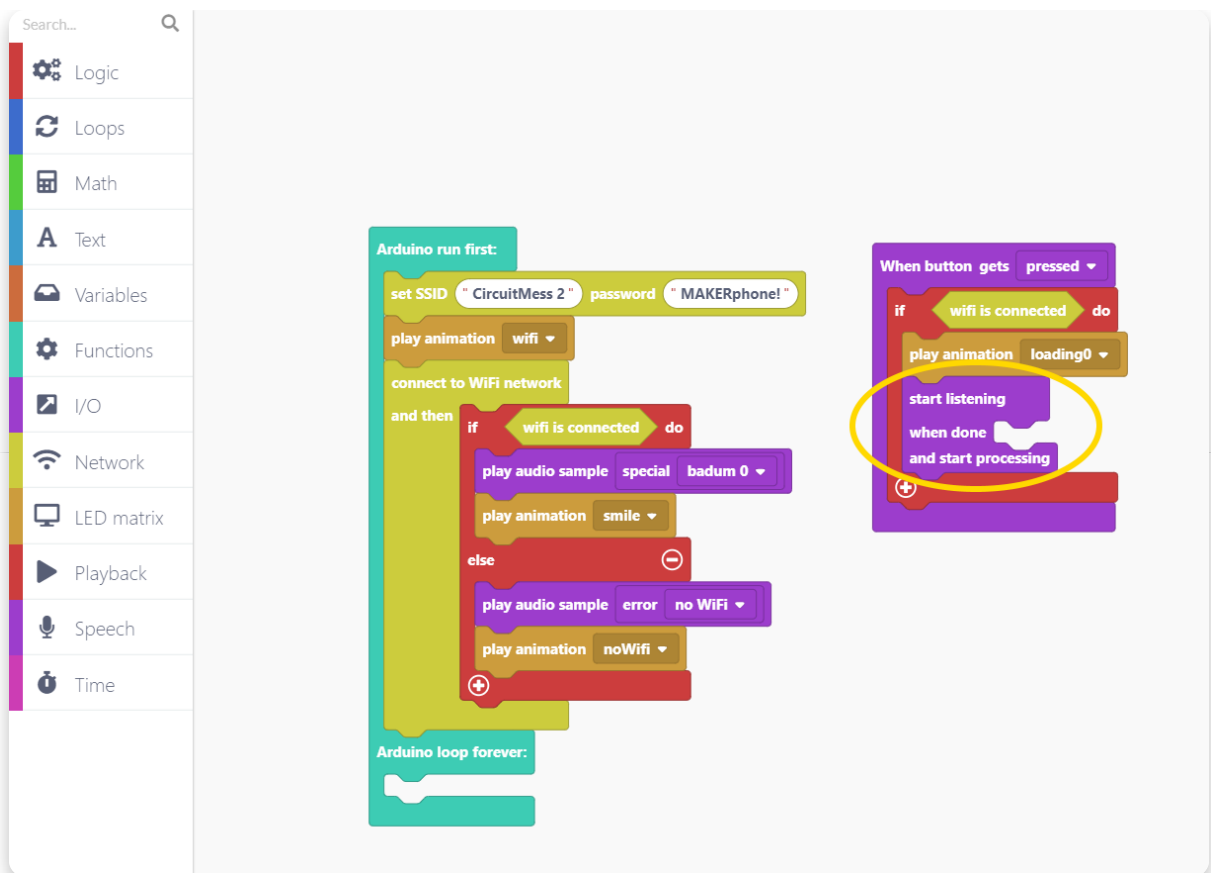
Pirmkārt, izdzēsiet bloku "synthesize and say", ar peles labo pogu noklikšķinot uz tā un nospiežot dzēšanas opciju.



Tagad mums vajadzēs "Start listening" bloku.

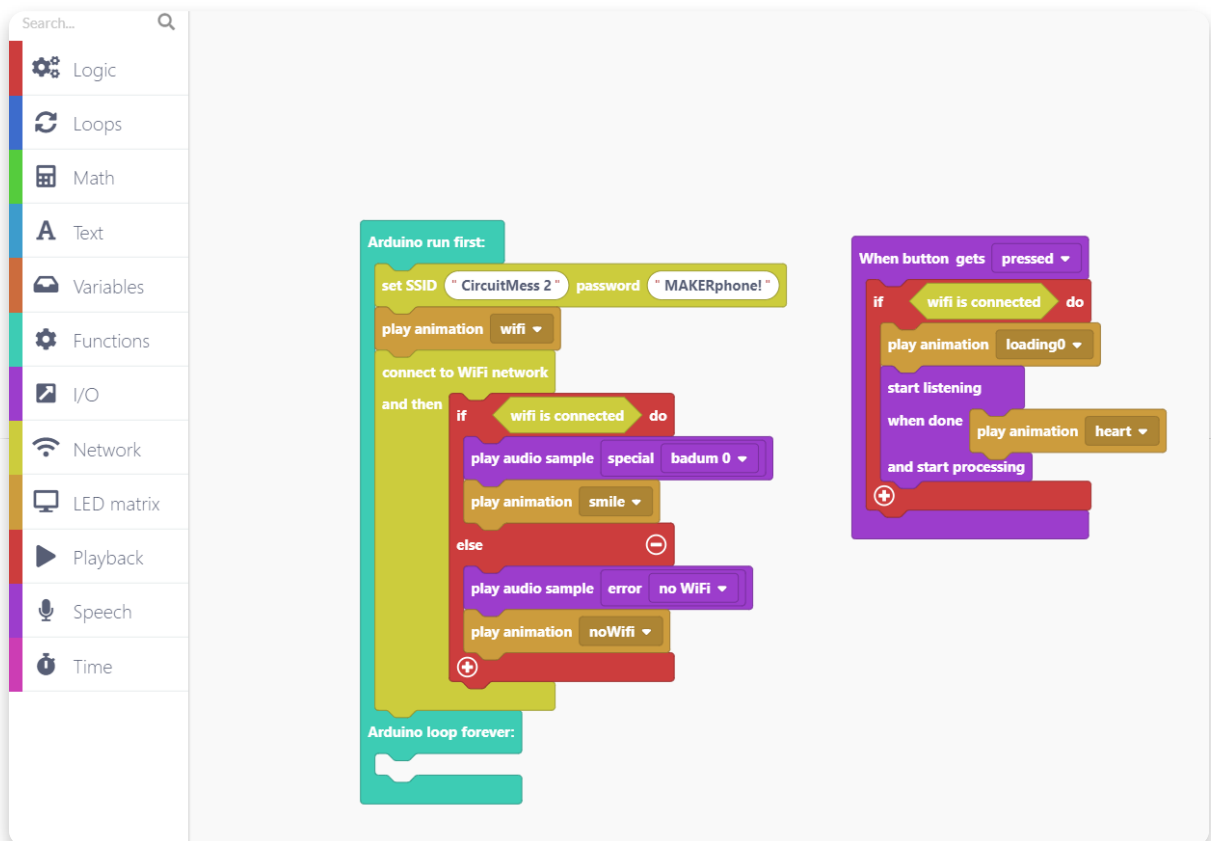


Novieto to šeit:

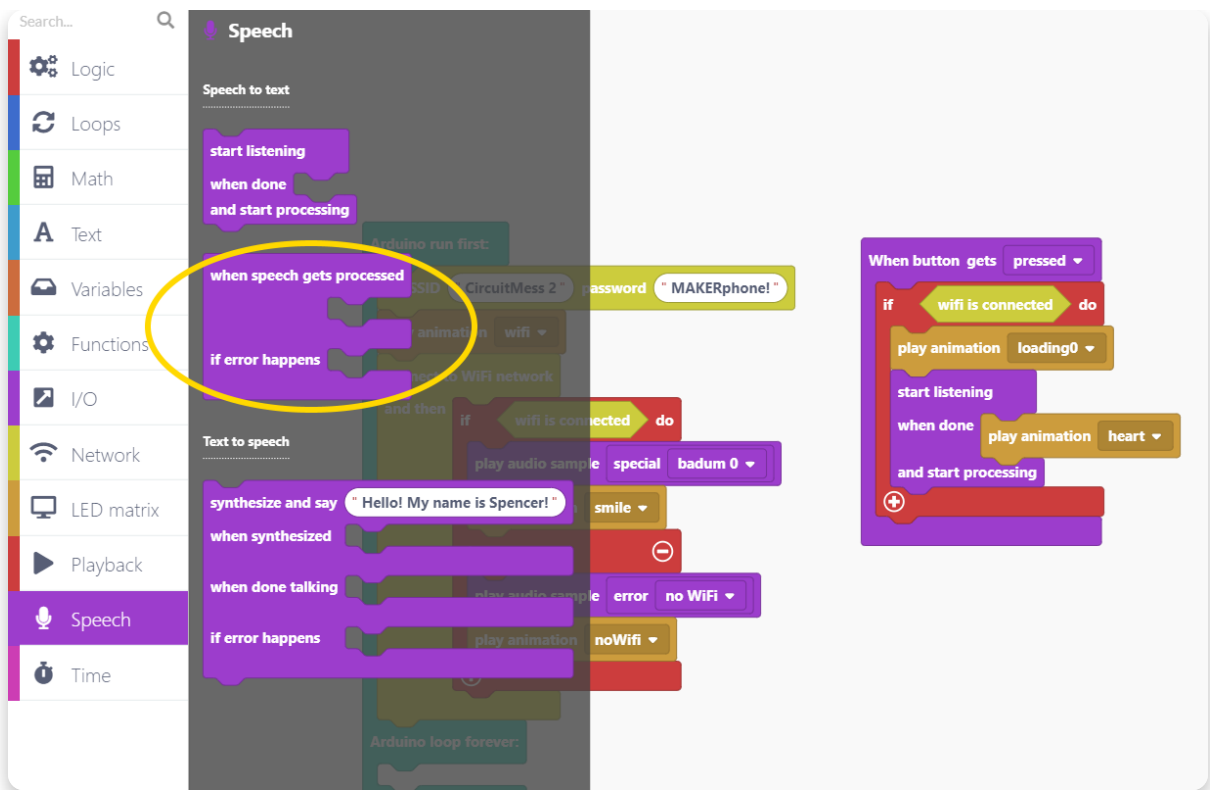


Izmantojot šo kodu, katru reizi, kad nospiežat tā sarkano pogu, Spenser pārbaudīs, vai ir izveidots savienojums ar WiFi, atskaņos ielādes animāciju un sāks klausīties jūsu teikto.

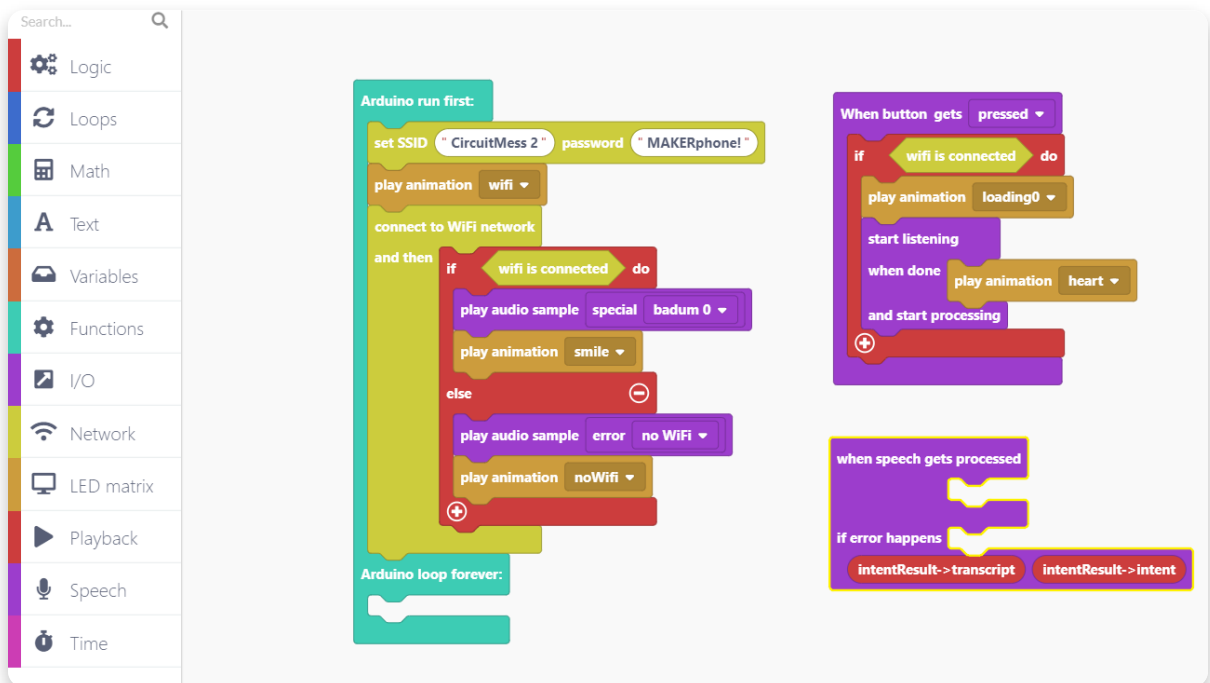
Kad Spensers ir pabeidzis klausīšanos, atskaņosim animāciju, lai mēs zinātu, ka tā tagad apstrādā mūsu balss komandu.



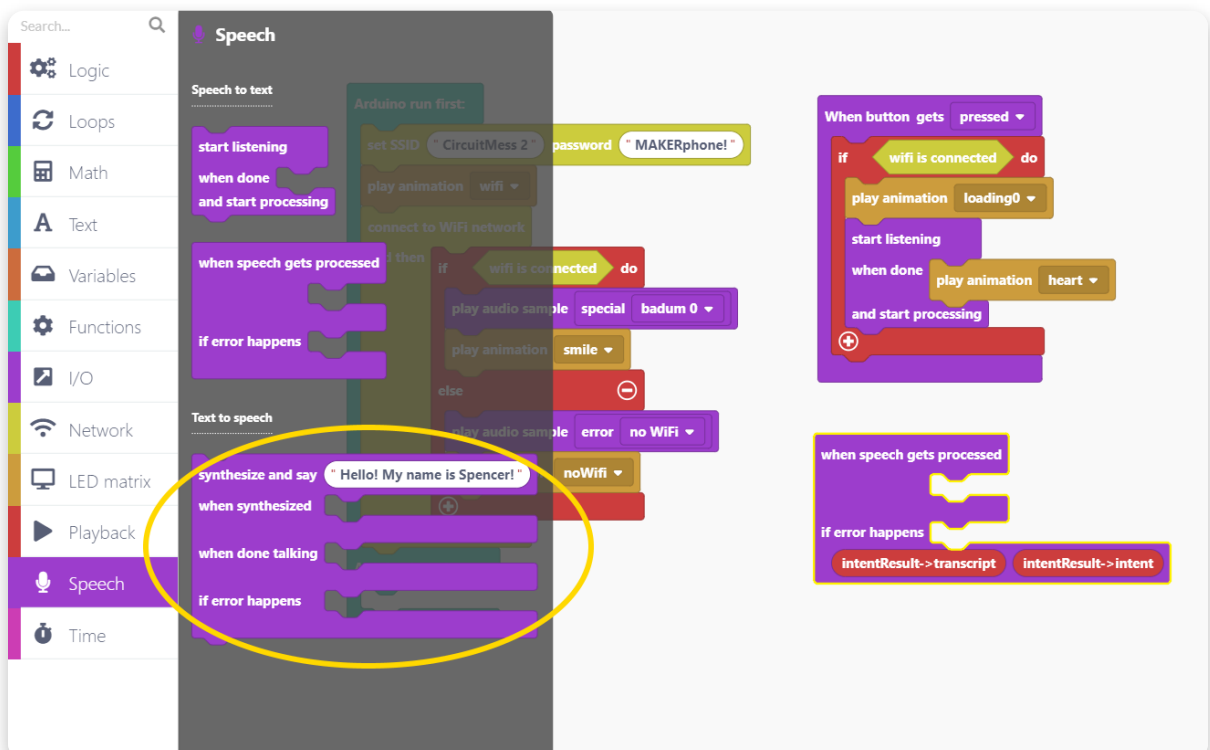
Tagad atrodiet notikumu bloku "when speech gets processed". Šis bloks tiks aktivizēts, kad Spensers apstrādās jūsu balss komandu un saņems stenogrammu no servera.



Turi un velc uz koda laukumu:



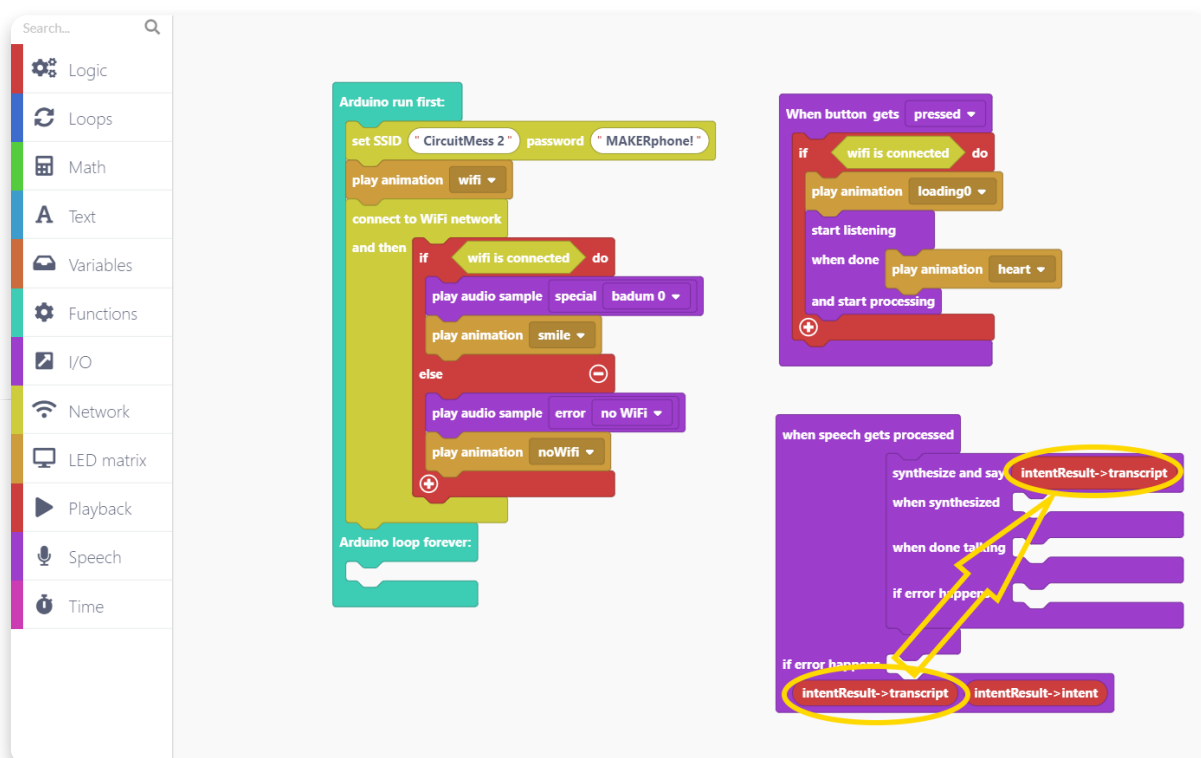
Tā kā mēs vēlamies, lai Spencers atkārtu to, ko esat viņam teicis, atrodiem bloku "Synthesize and say":



Novietojiet to šeit, jo mēs vēlamies, lai Spensers atkārtotu jūsu vārdus, tiklīdz jūsu balss ir apstrādāta:



Mēs vēlamies, lai Spensers saka tieši to, ko jūs teicāt. Tāpēc jums jāņem sarkanais vērtību bloks "intentResult-> transcript" un jānomet tas blokā "Synthesize and say".



Visbeidzot, atskaņosim runājošu animāciju, kad Spensera balss paraugs tiks generēts, un liks viņam pamirkšķināt, kad viņš būs beidzis runāt.



Lieliski, jūsu kods ir gatavs.

Saglabājiet to un palaidiet to savā Spencer.

Piesit Spencer galvu un pasaki viņam kaut ko (t.i., "Sveiks Spencer, tu esi mans draugs"). Spensers apstrādās jūsu balsis komandu un atkārtos to ar savu robotu.

Atjaunot Spencer versiju.

Atjaunot Spencer versiju.

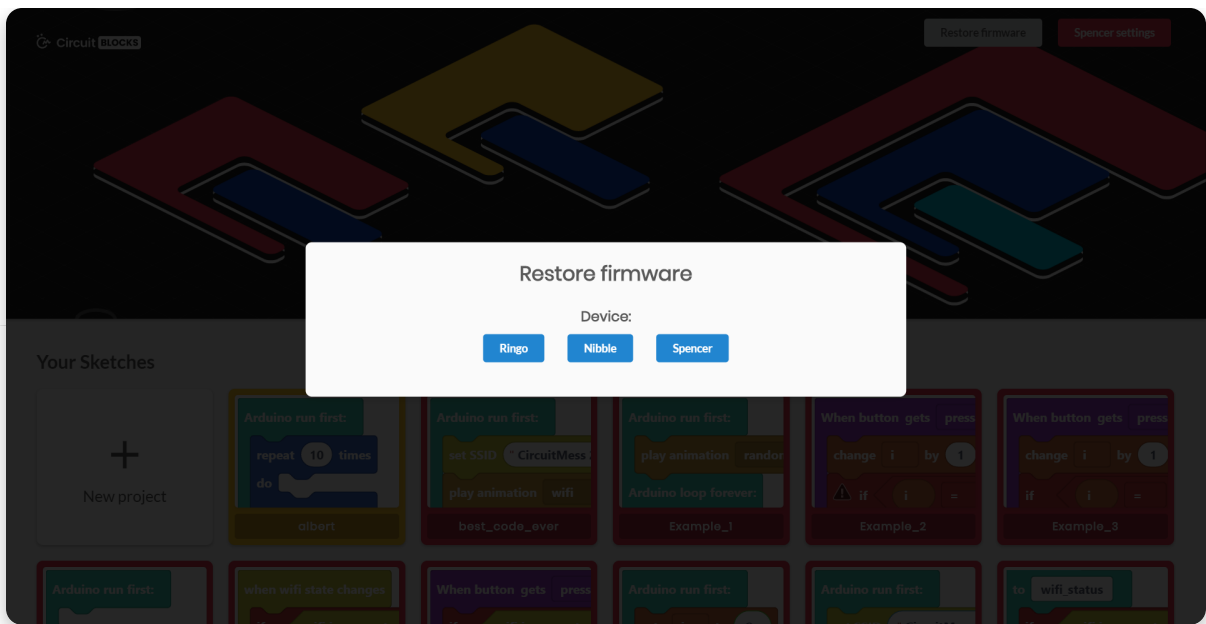
Kad esat pabeidzis kodēšanu un vienkārši vēlaties, lai jūsu Spensers atkal būtu "normāls", jums jāatjauno viņa bāzes programmaparatūra.

Tas ir pavisam vienkārši, vienkārši pievienojiet savu Spencer datora USB portam un augšējā labajā stūrī nospiediet pogu "Restore firmware".



Jums tiks piedāvāts logs, kurā jums jāizvēlas ierīce, kurai atjaunojat programmaparatūru.

Protams, izvēlieties Spenseru.



Pagaidiet dažas sekundes, un jūsu Spencers atkal darbosies kā parasti.

Jums tas jādara ikreiz, kad esat pabeidzis kodēt savu Spencer, ja vēlaties, lai viņš atgrieztos pie sākotnējās ārējās funkcionalitātes.

Ko tālāk ?

Jūs esat sasniedzis mūsu pirmās Spencer kodēšanas apmācības beigas, apsveicam!

Es ceru, ka jūs esat tikpat satraukti kā mēs par Spensera nākotni, jo nākamajos programmaparatūras un CircuitBlocks atjauninājumos mēs esam paredzējuši daudz uzlabojumus.

Tikmēr turpiniet patstāvīgi izpētīt un parādiet mums, ko esat paveicis ar savu Spencer, daloties uz CircuitMess kopienas forumu:

<https://community.circuitmess.com/>

Ja jums nepieciešama palīdzība saistībā ar ierīci, kā vienmēr, sazinieties ar mums pa **contacts@circuitmess.com**, un mēs palīdzēsim pēc iespējas ātrāk.

Paldies un turpini gatavot!

