Playground, Gizmos og Crumbs

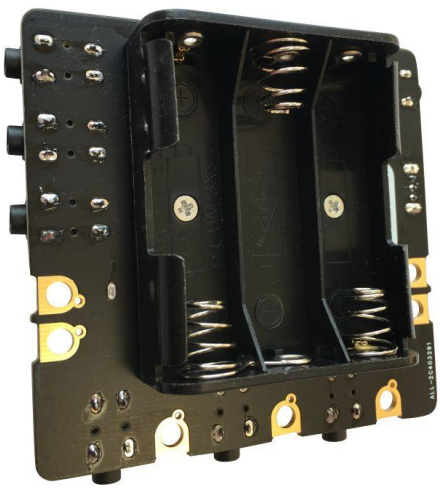
Med Playground fra 4Tronix får du en legeplads med komponenter (Gizmos), der er nemme at tilslutte med de medfølgende dc-stik. Playground er desuden bagud-kompatibel med de ældre ”Crumbs” fra Redfern.

Du kan programmere Micro:bit med www.makecode.microbit.org, hvilket du får en række eksempler på i denne guide. Du vil selvfølgelig kunne bruge andre sprog som f.eks. Python og Arduino til at kode din Micro:bit.

Playground

Forsiden:

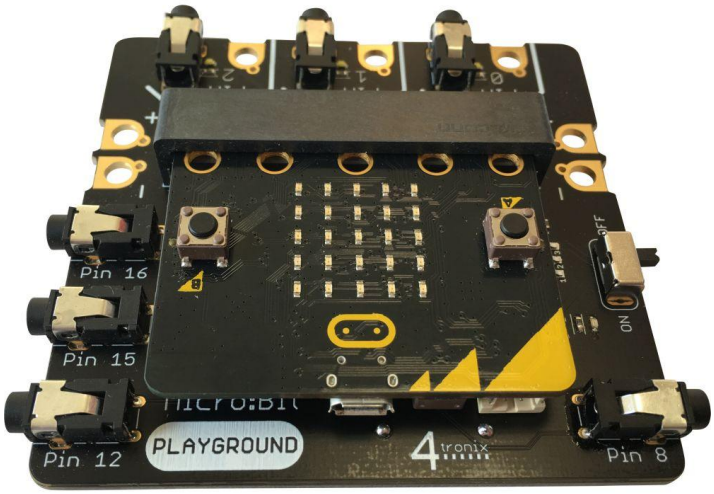
Bagsiden:



Micro:bit sættes ned i edge-connectoren på forsiden af Playground.
Micro:bit får strøm fra edge-connectoren (batterier på bagsiden), men du skal stadig bruge usb

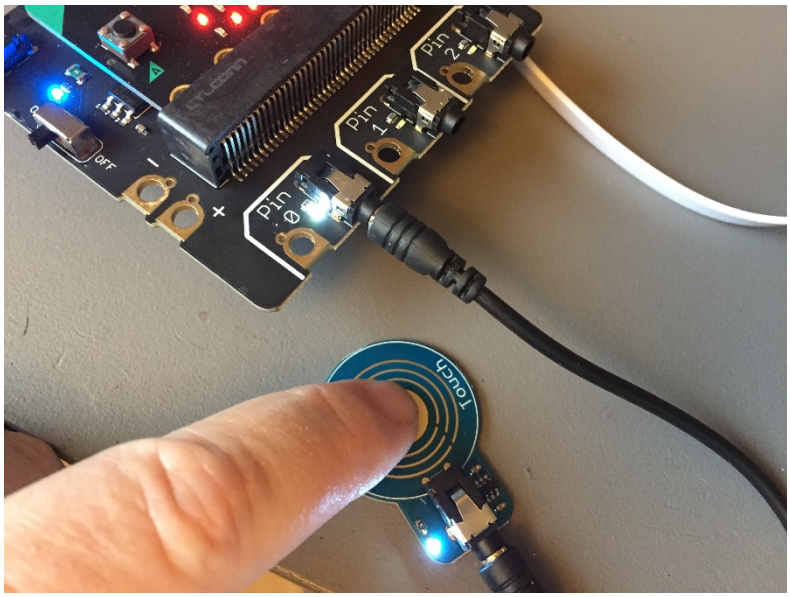
ladekablet til at downloade programmer til Micro:bit fra computer. Efter download kan du tage usb-stikket ud af Micro:bit og køre på batterier alene fra Playground.

Når batterierne er sat i holderen, kan du tænde on/off knappen. Ved ”on” lyser en blå LED på Playground lige ved kontakten.



Alt efter Gizmo og aktivitet vil en LED ved siden af dc-stikket på Pin 0,1 og 2 på Playground indikere status på Gizmo med et hvidt lys. (om der sendes eller modtages giver samme lys).

I dette eksempel modtages input fra Touch-sensoren på Pin0 på Playground. Når usb-kablet er koblet på Micro:bit, tænder LED uagtet om der er tændt for batterier. Men du får ikke lyd på f.eks. buzzer uden at kontakten er tændt.

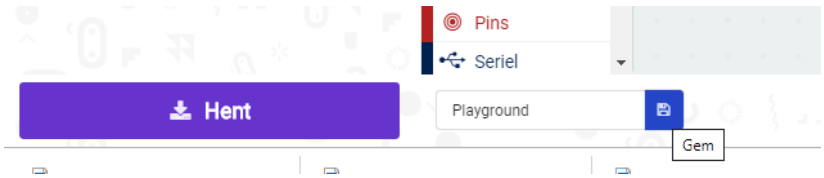


Download programmer til Micro:bit.

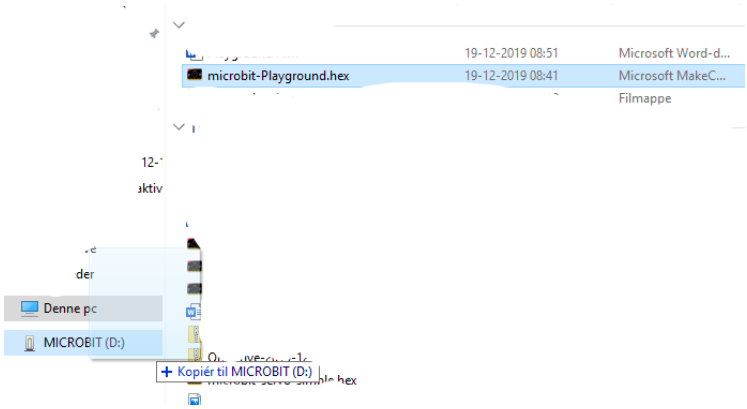
Inden vi går i gang med de enkelte Gizmos (og senere Crumbs) får du lige en hurtig opfrisker til at lægge programmer over på Micro:bit.

Læg koden over på Micro:bit fra computer

Sæt usb-kablet til Micro:bit og computer.
Klik på "Hent" eller på save-ikonet ved feltet, hvor du navngiver din fil.



Find filen på dit drev og træk den over på Micro:bit. Se næste side.



(Arbejder du med pair device eller på tablet/iphone, kan du koble til Micro:bit via bluetooth. https://support.microbit.org/support/solutions/articles/19000051025-pairing-and-flashing-code-via-bluetooth*)*

Det kan være en god ide at lave en pair device også fra computeren, hvis der skal testes mange koder. Dette kan gøres med usb-ladekablet under tandhjulet oppe i højre hjørne af editoren og vælge pair device.

Gizmos

Gizmos er forskellige komponenter med f.eks. sensorer, der kan forbindes til Playground.
Gizmos forbindes til Playground med dc-stik. I Super Kit følger der dc-stik med kabler i forskellige længder.

Tjek at stikket er trykket helt på plads på både Playground og Gizmo – især hvis ikke Gizmo virker korrekt er det ofte stikket, der ikke er trykket rigtigt ind.
– eller kontakten på Playground er ikke tændt, hvilket også vil begrænse hvad du får ud af Gizmos.

Gizmos er inddelt i 5 farvekategorier
– bemærk at ikke alle Gizmos er med i startsættet.

- Blå: Digital input (trykknap og berøringssensor)
- Gul: Analog input (lyssensor og potentiometer)
- Sort: Digital output (LED og buzzer)

- Rød: Analog output (Servo og Motor)
- Hvid: Smart Pixels (aka “neopixels” or “sparkles”), individuelt adresserbare RGB-pixels, der kan sættes i forlængelse af hinanden. (Flame, Blaze og Fire Stick)

Inddelingen i farver kan hjælpe dig til at bruge de rigtige klodser i f.eks. PINS-biblioteket på Makecode.

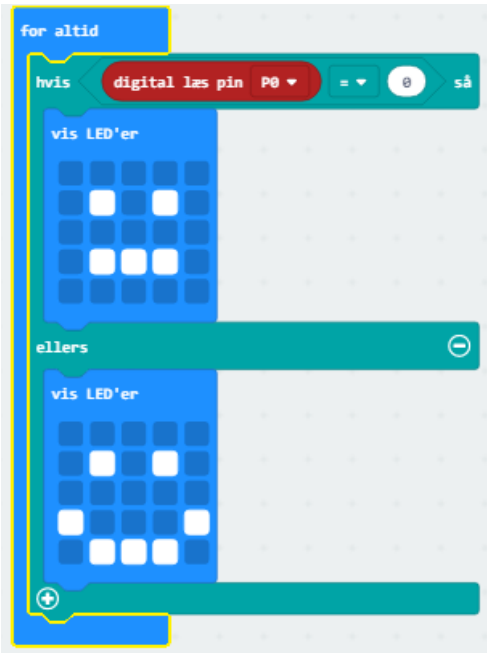
I det følgende får du nogle kodeeksempler inspireret af 4Tronix hjemmeside og opdateret til nyeste editor på Makecode. I de originale guides på 4Tronix ser du også eksempler på programmering med Python. https://4tronix.co.uk/blog/?p=1425

Blå Gizmo – Digitalt Input

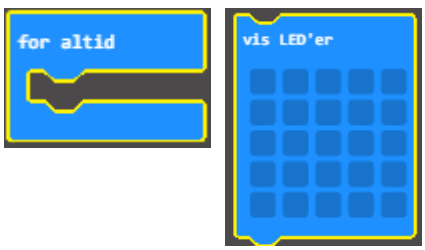
Til blå Gizmos kan du bruge ”digital læs pin” til at fange, hvad der sker ude på Gizmoen.
Du kan bruge koden nedenfor til at prøve det af på Touch Gizmo. Der returneres en digital værdi enten er der tændt eller slukket, on/off, 1 eller 0, High eller low.

Gå på www.makecode.microbit.org og start et nyt projekt op.

Find de klodser, som dem du ser i koden nedenfor, og byg koden op. Du får lidt hjælp på næste side.



Du skal bruge en ”for altid” som er på skærmen, når du starter editoren op. Ellers kan du finde den i biblioteket GRUNDLÆGGENDE. Du skal også bruge en ”vis LED’er” fra samme bibliotek.



Fra LOGIK biblioteket henter du 2 klodser ind: ”hvis..ellers” og en sammenligningsklods.

Fra PINS henter du en ”digital læs pin”.

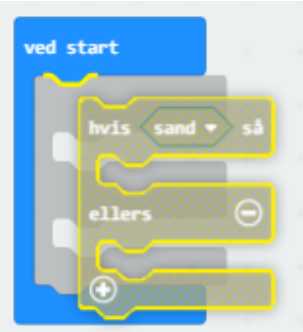
Er det første gang du leger med klodser fra LOGIK eller PINS, kender du måske endnu ikke tricket med at sætte klodser ind i hinanden. Dette hedder at ”indlejre” når vi programmerer.

I koden bruger du ”hvis…ellers” til at sætte en betingelse op for, hvornår der skal ske noget på Micro:bit. I eksemplet vil vi vise en mellemfornøjet smiley, så længe der **ikke** trykkes på gizmoen. Når der trykkes, vises en glad smiley så længe du holder trykket på gizmoen.

Så vi skal sammenligne 2 scenarier. Når der trykkes og når der IKKE trykkes. Det bruger vi sammenligningsblokken til og den røde ”digital læs pin” til.

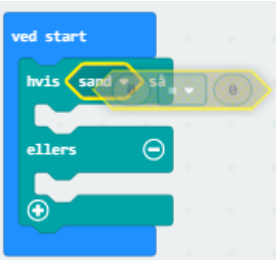
Sæt nu klodserne sammen sådan:

Træk den grønne(grå) ”hvis…ellers” hen i ”for altid” blokken.

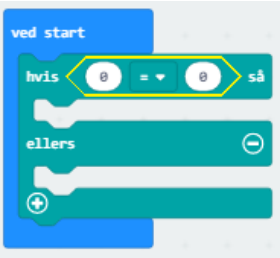


>>>>>

Gør det samme med sammenligningsklodsen til feltet ”sand”.

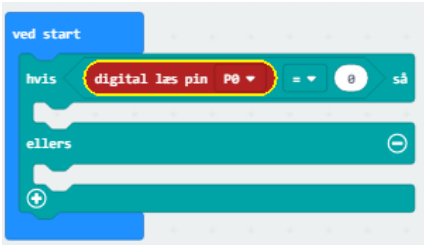


>>>>>

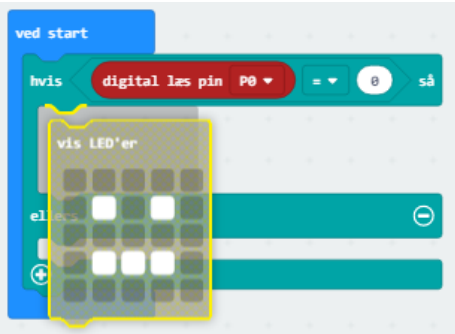


Nu skal du bruge den røde ”digital læs pin” klods. St den ind ved det første ”0” i sammenligningsklodsen.

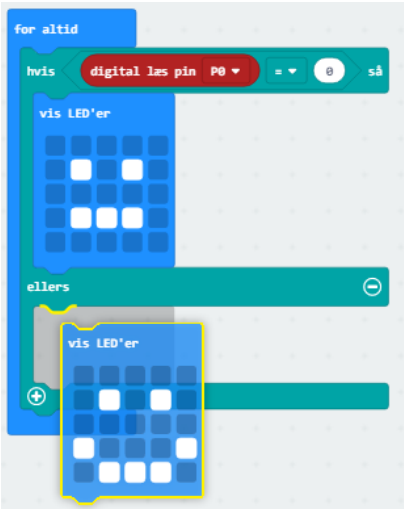
>>>>>>



Til sidst skal du angive, hvad der skal ske, når du trykker og når du ikke trykker. I eksemplet bruger vi 2 ”vis LED’er” klodser til at angive 2 forskellige smileys og sætte dem ind i hvert scenarie.



>>>>>>



Denne kode vil kunne bruges på alle de blå (også fremtidige) Gizmos.

Forbind Gizmo

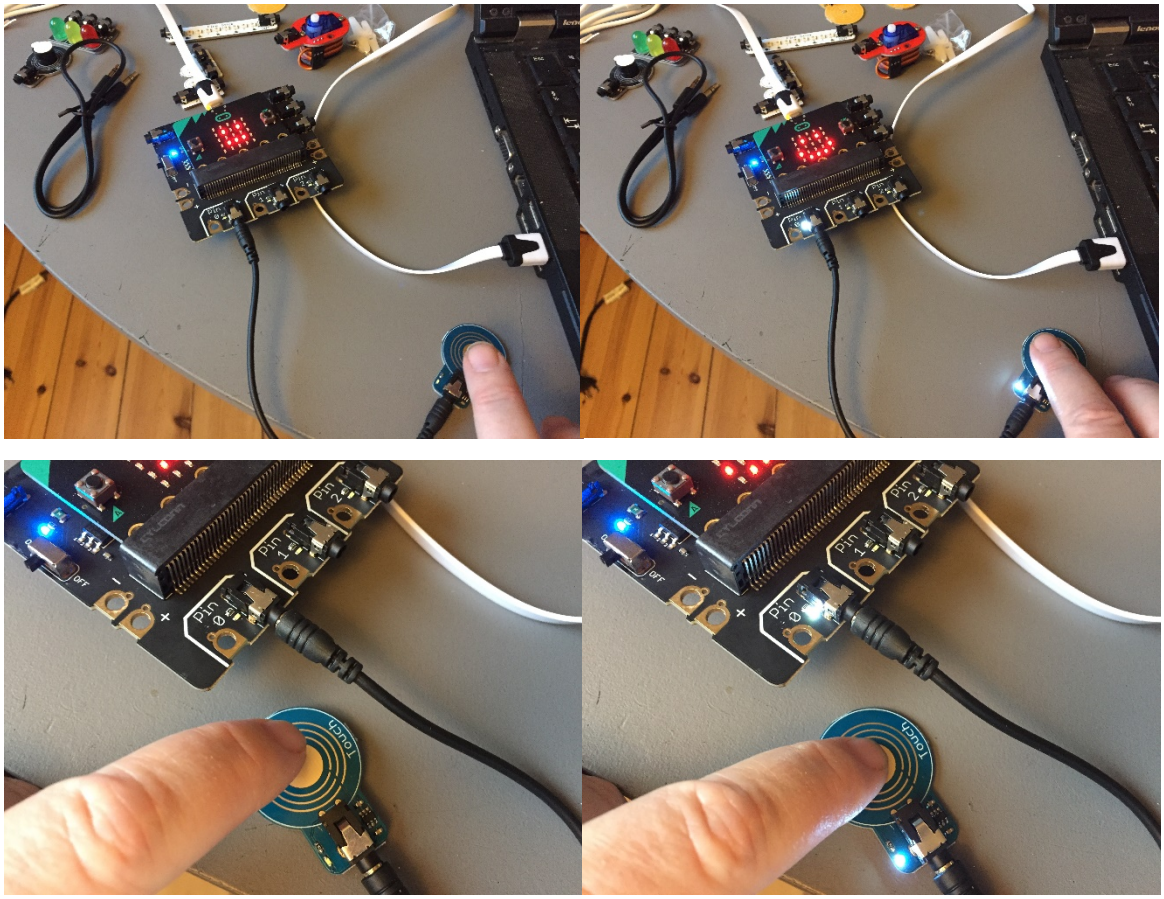
I koden bruger vi P0 i den røde klods. *Så du skal forbinde Gizmo’en til Pin0 på Playground*.

1. Sluk for batterierne på Playground (sæt knappen til ”off”)

2. Forbind Gizmo’en ”Touch” til Pin0 med dc-stik.

3. Tænd for batterierne på Playground.

Test din kode. Se f.eks. om LED ved stikket på Playground tænder og slukker når du skiftevis rører og ikke rører ved fladen på Gizmo’en. Smiley på Micro:bits display skal også helst ændre sig i takt med at du rører og slipper.



Gul Gizmo – Analogt Input

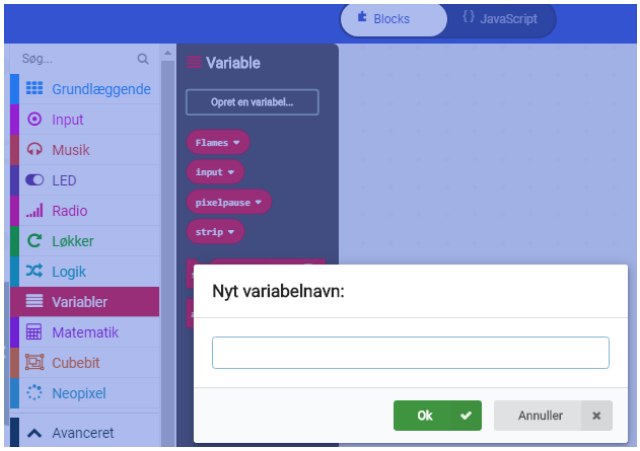
Gule Gizmos giver dig et analogt input. I modsætning til de blå Gizmos har du mange punkter mellem tændt og slukket. Et eksempel kan være at du vil måle dagslyset. Mellem *helt lyst* og *helt mørkt* er der mange små punkter, hvor der er lidt mindre mørkt eller lidt mere lyst. Lyset slukkes og tændes ikke som en lampe.

Vi bruger stadig klodser fra PINS biblioteket, men denne gang bruger vi *analoge klodser*.

Dette kode-eksempel kan du bruge til både Light og Dial Gizmos.

Du måler hvor meget strøm mellem 0 (0Volt) og 1023 (3Volt) der lukkes ind gennem den Gizmo du bruger. Forskellen er at du med Dial Gizmo selv skruer op og ned og at lyset bestemmer for Light Gizmo.

Lav en variabel du kalder input i biblioteket VARIABEL



En variabel er en værdi, der kan ændre sig. De fleste kender til scoren i et spil, der tæller point. Scoren er et eksempel på en variabel, der forandrer sig gennem spillet.
I eksemplet nedenfor er ”input” en variabel, der forandres alt efter hvor meget strøm, der kommer igennem en gul Gizmo. Du kan kalde den lige det du vil; gulerod, bilpark osv. Men det er en god ide at bruge navne, der siger noget om, hvad variablen skal bruges til. Du kan f.eks. derfor i eksemplet kalde den strøm eller input. Vi måler jo på f.eks. lysniveauet ændrer sig.

De øvrige klodser, du skal bruge, finder du i GRUNDLÆGGENDE.

0. Gem koden og læg den over på Micro:bit.
1. Sluk Playground med on/off knappen.
2. Forbind Dial Gizmo.
3. Tænd så Playground igen.

Prøv nu at skrue op og ned på drejeknappen. Læg mærke til hvordan tallene på Micro:bit ændrer sig.

(OBS. Evt. skal du først sætte hætten på den lille drejeknap. Drej den helt ud til en af siderne og sæt hætten på med spidsen ud mod stoppunktet.)

Prøv den samme kode med en Light Gizmo.

Synes du tallene er lidt uoverskuelige, kan du udvide koden en smule og få mere enkle tal.

Fra MATEMATIK henter du en divisions-klods og dividerer input fra pin med 100. Sæt ”analog læs pin” ind ved det første 0 og skriv 100 i det andet felt med 0.

Sæt så hele klodsen ind i ”sæt ……. til”:

I ”vis nummer” klodsen er du nødt til at sætte en ”afrund” fra MATEMATIK ind i feltet, så du får et helt tal ud.

Læg koden over på Micro:bit. Nu skulle du gerne få et tal mellem 0 og 9, når du tester igen med de gule Gizmos.

Sort Gizmo – Digitalt Output

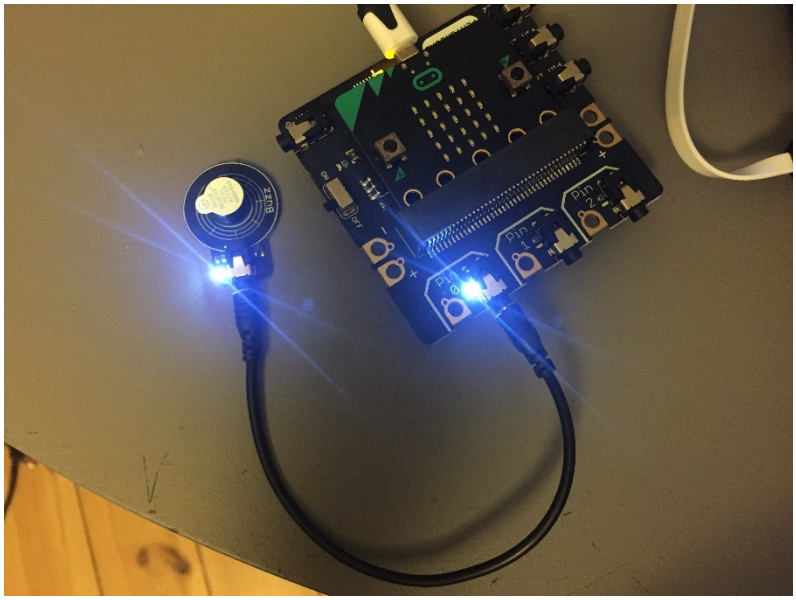
Med ”digital skriv til” klodser i PINS biblioteket kan du lave lyd med din Buzzer Gizmo eller et trafiklys til din Trafic Gizmo.

Til Buzzer Gizmo kan du bruge enten klodser fra MUSIK biblioteket eller du kan bruge ”digital skriv pin” fra PINS biblioteket.

Først et eksempel med PINS

Find selv klodserne i GRUNDLÆGGENDE og PINS og byg koden op i Makecode.

0. Gem koden og læg den over på Micro:bit.
1. Sluk Playground med on/off knappen.
2. Forbind Dial Gizmo.
3. Tænd så Playground igen.



Når Buzzer Gizmo er forbundet til Pin0, kan vi også kode den til at spille en lille melodi fra MUSIK biblioteket. Buzzeren giver ikke nogen stor fin lyd til musik, men det kan alligevel være lidt sjovt at prøve.

I næste eksempel skal du bruge klodser fra INPUT og MUSIK.

”når der trykkes på knappen A” finder du i INPUT. Hent den ind og ret den til ”B”. Hent den så ind igen og lad den nye være knap A.

Hent en ”start melodi”ind og sæt den i knap A. Vælg en af melodierne ved at klikke på pilen i midten. Du kan høre den på simulatoren, hvis lyden er slået til på din computer.

Gentag det samme for knap B og vælg en anden melodi.

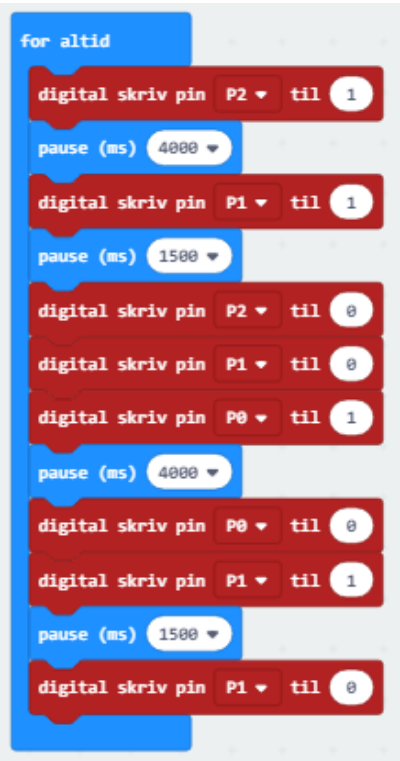
Gem koden og læg den over på Micro:bit og test at du får to forskellige melodier alt efter om du trykker på knap A eller B. Leg selv videre med f.eks. kode dine egne melodistumper.

Kodeeksempel til Trafic Gizmo.
Du skal bruge 3 ledninger med stik og forbinde hver pin med hver sin LED på Gizmo'en.

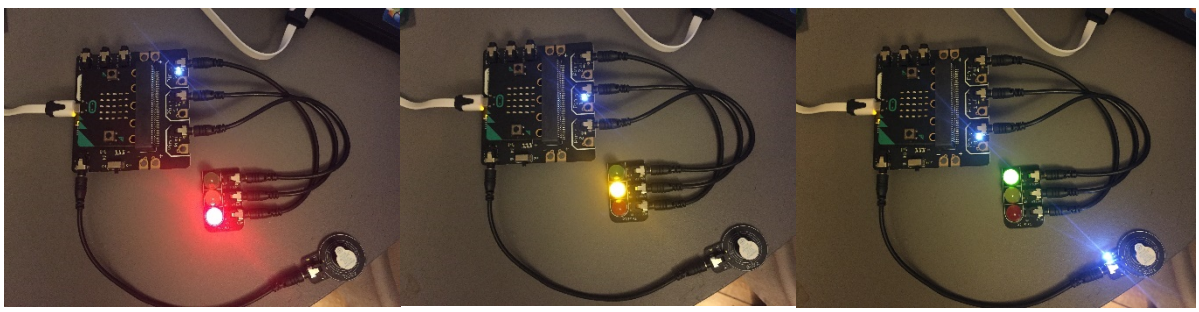
Koden du skal lave, er meget enkel. Men du skal alligevel tænke dig om og holde godt styr på de tre pins, der bruges. De tre LED lyser ikke lige længe og rød og gul lyser nogle gange på samme tid, mens grøn altid lyser alene. Prøv selv at lægge mærke til det næste gang, du står ved et lysreguleret kryds.

Inden du begynder at bygge koden nedenfor, kan du se den igennem og se lyset for dig som det skifter. Byg så koden og se om du får ret.

Tip
Læg mærke til at tallet i "pause (ms)" er nogle andre end dem du kan vælge ved at klikke på pilen. Du kan klikke med musen i feltet og skrive lige det heltal du vil, så det passer til din kode.



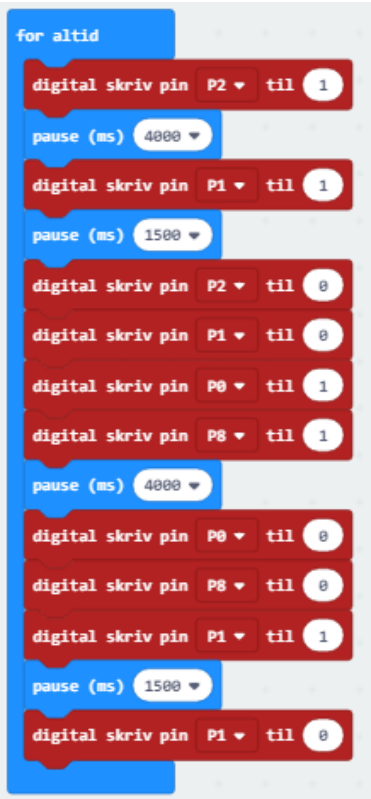
0. Gem koden og læg den over på Micro:bit.
1. Sluk Playground med on/off knappen.
2. Forbind Dial Gizmo.
3. Tænd så Playground igen.



Under du dig over at Buzzer Gizmo er med på billederne ovenfor?
I mange lyskryds er der en lyd, så svagtseende kan ”høre” om der er rødt eller grønt.

Du kan bruge din Buzzer Gizmo sammen med Trafic Gizmo og lave en hyletone når der er grønt.
I eksemplet forbindes Buzzer Gizmo til Pin8 på Playground. Pin8 er digital. Eksperimentér selv med de øvrige pins 12, 15 og 16 om de kan bruges på samme måde.

Du skal ændre koden fra før en lille smule… Se på næste side; kan du se forskellen? Udvid din kode, gem den og læg den over på Micro:bit.



OBS! Husk at tænde kontakten på Playground for at få lyd i Buzzeren.

Rød Gizmo – Analogt Output

I den røde serie af Gizmos kan kun Servo-motoren pt. anvendes til Micro:bit.

Vi starter med at bruge/hente ”ved start” klodsen fra ”GRUNDLÆGGENDE”. Fra samme bibliotek kan du også hente ”pause (ms)..”

Til servoen kan du starte med at bruge klodsen ”servo skriv pin” fra biblioteket PINS

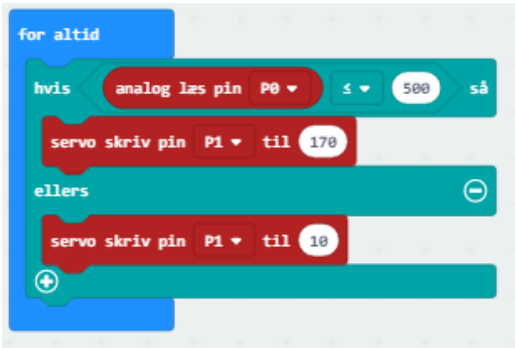
Behold i eksemplet P0.

Du skal bruge nogle stykker af klodsen og rette gradtallet (180) til forskellige positioner.

Nu har du en opstartsrutine, der viser at servoen kan dreje i begge retninger.

Herfra kan du bruge andre events til at få servoen til at svinge ud. F.eks. kunne du kombinere med f.eks. input fra gule Gizmos til få Servoen til at svinge.

Prøv at lave følgende kode i samme program som den du lige har lavet.

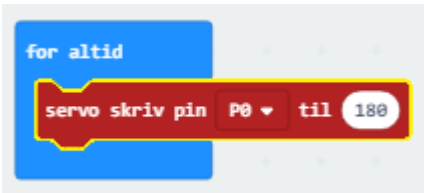


Læg koden over på Micro:bit og tilføj en Light Gizmo i Pin0 på Playground.

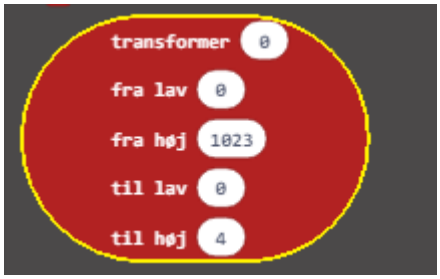
Vil du prøve at lege på samme vis med Dial Gizmo kan du forsøge dig med denne kode, hvor du ”mapper” input fra Dial Gizmo med Servoen.

Lav et nyt projekt med samme startkode og lav så en ny kode med ”for altid”.

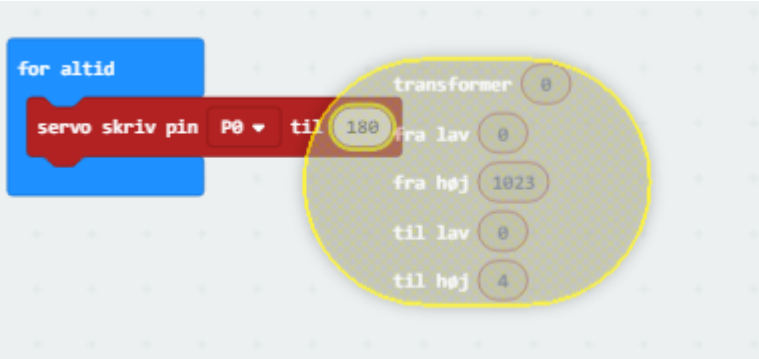
Hent ”for altid…” ind fra GRUNDLÆGGENDE og hent en ”servo skriv pin ..” ind og sæt dem sammen.



Find nu denne klods inde i biblioteket PINS.

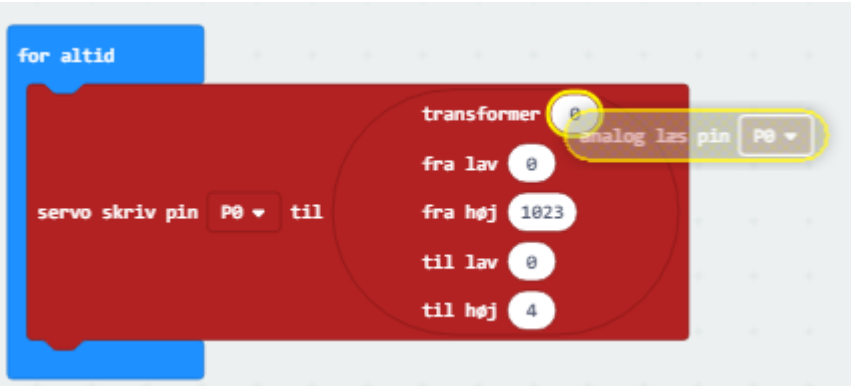


Sæt den runde klods ind i ”servo skriv pin” i feltet med ”180”.



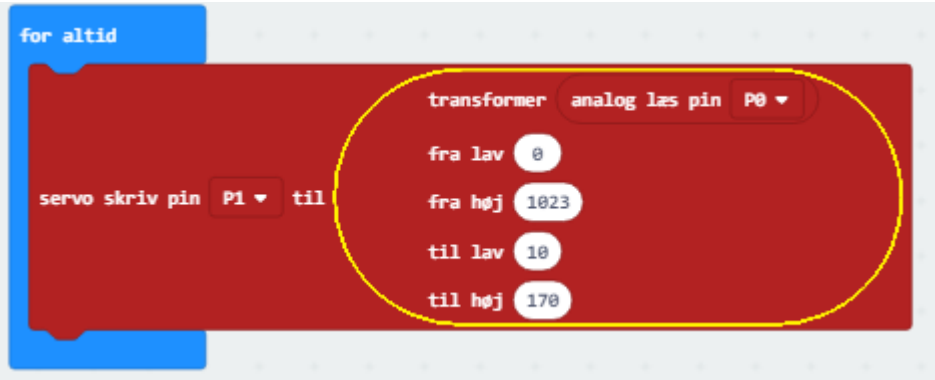
Nu skal du lige have rettet værdierne i den runde klods lidt til.

Først sætter du klodsen ”analog læs pin P0” ind i feltet ”transformer”



Nu skal du rette tallene i de sidste to felter (0 og 4 rettes til 10 og 170).

Ret også P0 i ”servo skriv pin..” til P1.

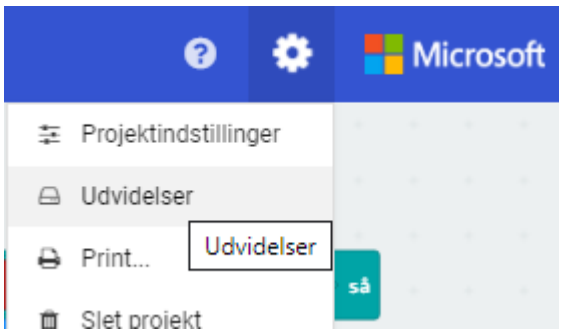


Læg koden over på Micro:bit og sæt Dial Gizmo i Pin0 på Playground

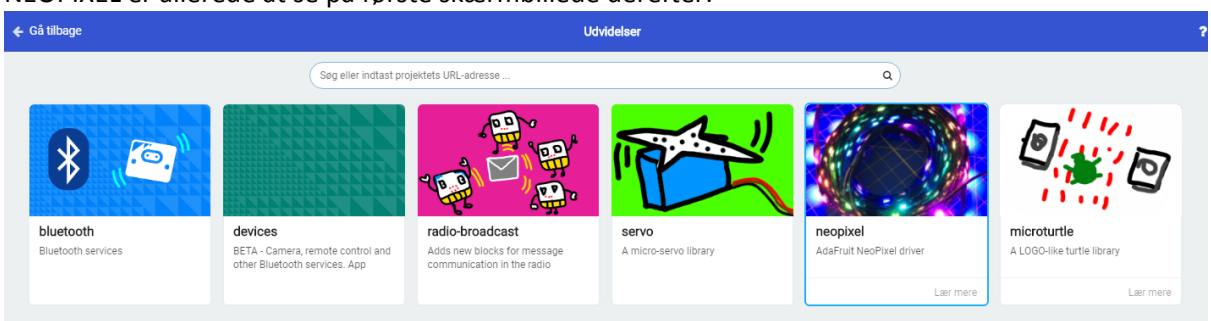
Hvid Gizmo – Smarte Pixels

Til de hvide Gizmos skal du bruge en udvidelse til Makecode: NEOPIXEL.

Du kan finde udvidelser ved at klikke på tandhjulet oppe i højre hjørne på editoren i Makecode og vælge "Udvidelser".



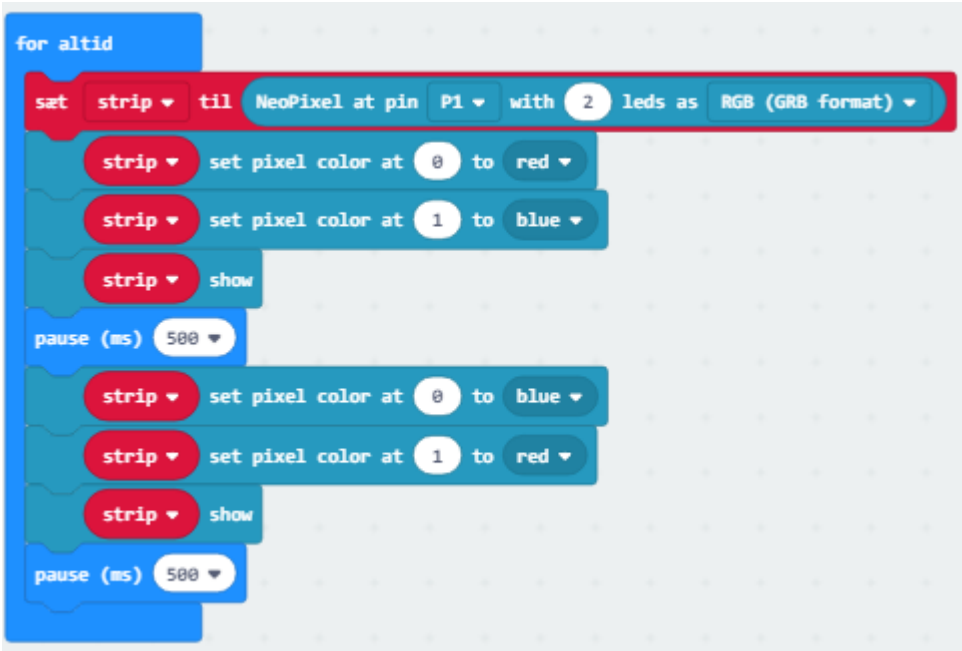
NEOPIXEL er allerede at se på første skærmbillede derefter:



Klik på Neopixel ikonet.

Tilbage i editoren kan du nu se at der er kommet en kategori mere i biblioteket.

Kode-eksempel til 2 Flame Gizmos



I eksemplet bruger du kun klodser fra GRUNDLÆGGENDE og NEOPIXEL.

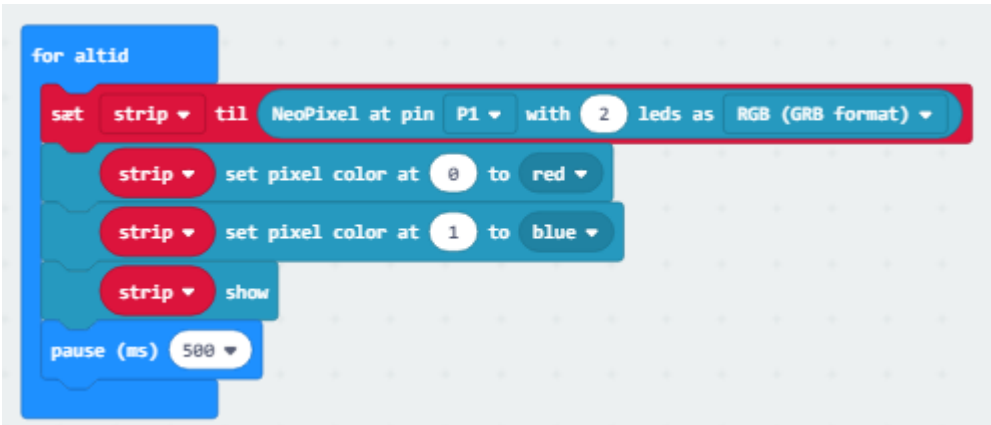
Med ”sæt strip til…” definerer vi, hvor mange NEOPIXELS vi har på samme streng. (strip) og hvilken pin vi vil forbinde vores NEOPIXELS til. I eksemplet bruger vi P1(pin1) og angiver de 2 Flame Gizmos.

Herefter går vi videre til at definere, hvilke farver henholdsvis den ene og den anden Flame Gizmo skal lyse med. Du har 16 millioner farver at lege med, men i eksemplet her leger vi med 2 farver. Målet er at få de to Flame Gizmos til skiftevis at lyse rødt og blåt modsat af hinanden.

Bemærk at når du tæller NEOPIXELS, tæller du fra og med 0. Så de to Flame Gizmos hedder henholdsvis ”0” og ”1”.

Hvis du ser på klodserne i koden, kan du måske genkende at der er to sæt med næsten samme kode.

Start med at lave første del af koden ved at finde klodserne i NEOPIXEL (og GRUNDLÆGGENDE) og sæt dem sammen som på næste side. Ret de steder hvor der står noget andet hos dig.

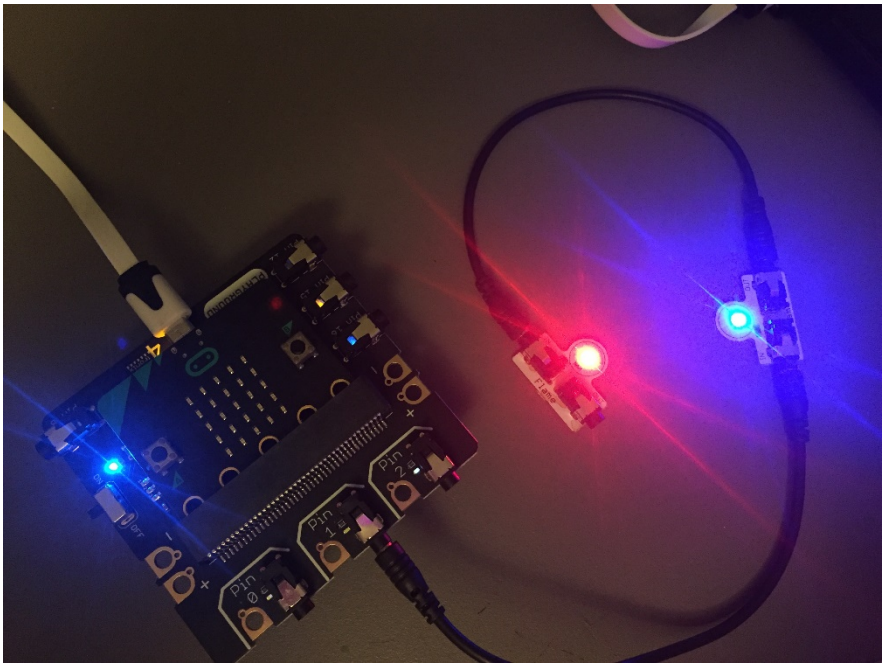


”strip set pixel color at…” bestemmer farven på de 2 Flame Gizmos. I koden er de sat til rød og blå.

”strip show” skal sættes ind for at aktivere koden på de 2 Flame Gizmos.

”pause (ms)…” er ikke så vigtig i den første del, men vil få stor betydning, når der tilføjes andre farve i koden.

Læg koden over på Micro:bit og forbind de to Flame Gizmos til Pin1 på Playground.



Der sker nok ikke andet end at hver Flame Gizmo lyser med hver sin farve.
For at få dem til at blinke, laver vi en blok mere med samme 2 ”strip set pixel color at…” en ”strip show”og ”pause..” og lægger ind efter den første pause:

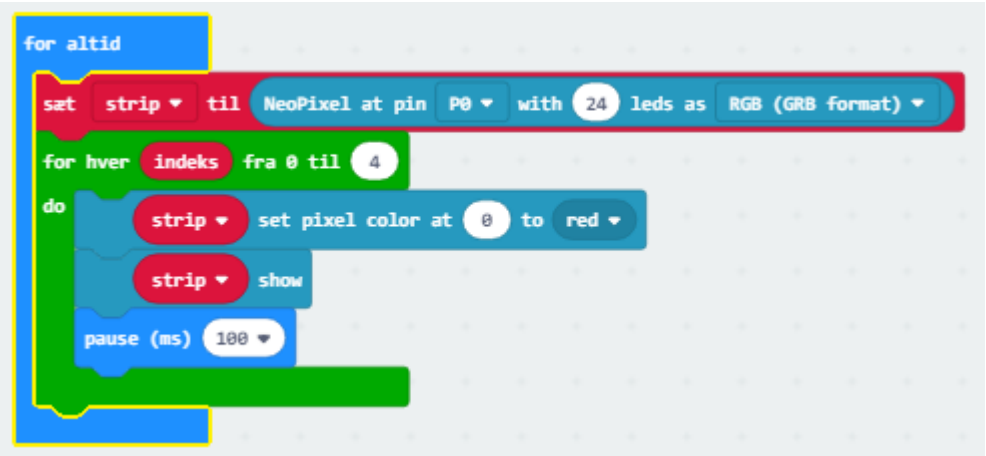
Læg mærke til at farverne i den nye blok i eksemplet er vendt om. Så 0 nu lyser blåt og 1 lyser rødt. Gør det samme i dit eget eksempel og læg igen koden over på Micro:bit. Nu skulle se to Flame Gizmos gerne blinke blåt og rødt.

Leg selv videre med andre farverkombinationer og sæt f.eks. flere farver på.

Kode til Fire Stick Gizmo

Fire Stick Gizmo kan bruges alene eller i en streng sammen med de 2 Flames Gizmo og kodes på samme måde som de 2 Flames. Forskellen er at på Fire Stick har du 8 NeoPixels at holde styr på.

Først samler vi alle klodser til koden og sætter dem sammen:



Når du har en kode som ovenfor, retter du den til:

Koden i mindre bidder

Først definerer vi en streng af NeoPixels i ”sæt strip til…” Ret P0 til P2 og ret 24 til 8.

Løkken ”for hver index fra 0 til 7”

Løkker finder du inde i biblioteket: LØKKER. Løkken vi bruger her, har sin egen variabel ”index” som du kan bruge i koden. Tallet 4 i sidste felt ændrer du til 7.

Vi har 8 pixels, men tæller fra og med 0. Derfor ændrer vi til tallet 7 (ikke 8). Løkken vil gentage alt, hvad der ligger inde ved ”do” for hver pixel (i alt 8).

Koden inde i løkken

Til koden inde i løkken retter du i ”set color at..” så variablen ”index” sættes ind i stedet for 0. Du finder den runde ”index” inde i biblioteket VARIABLER.

I samme klods retter du også farven, hvis du vil have en anden farve end rød. Klik på pilen i feltet hvor der står ”red”.

Variablen ”Stick”

Du laver en ny variabel som i eksemplet her kaldes ”Stick”.

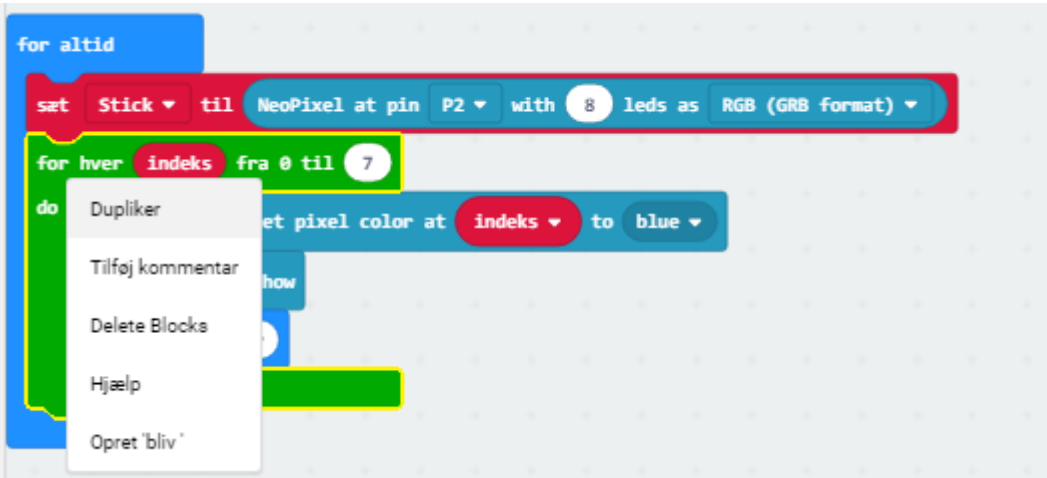
Når variablen er lavet, skal du alle steder i din kode, hvor der står ”strip” ændre det til ”Stick” ved at klikke på pilen ved siden af ”strip”.

Ved at gøre dette kan du have både din Fire Stick Gizmo og dine Flame Gizmos til at køre på samme tid i hver deres pin på Playground. Hvis de begge hedder ”strip” konflikter de, og koden vil ikke virke korrekt.

Nu skulle din kode gerne se sådan ud:

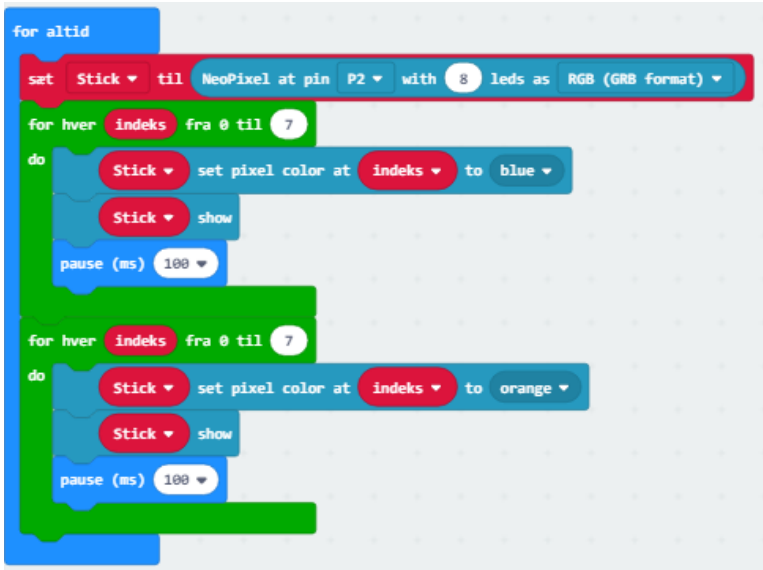
0. Gem koden og læg den over på Micro:bit.
1. Sluk Playground med on/off knappen.
2. Forbind Fire Stick Gizmo.
3. Tænd så Playground igen.

Du kan nu prøve at kopiere (med dupliker) hele den grønne løkke og sætte den ind efter den første løkke.



I den nye løkke laver vi nogle ændringer.

Først retter vi farven i ”stick set color…” til en anden farve. I eksemplet er der valgt orange.



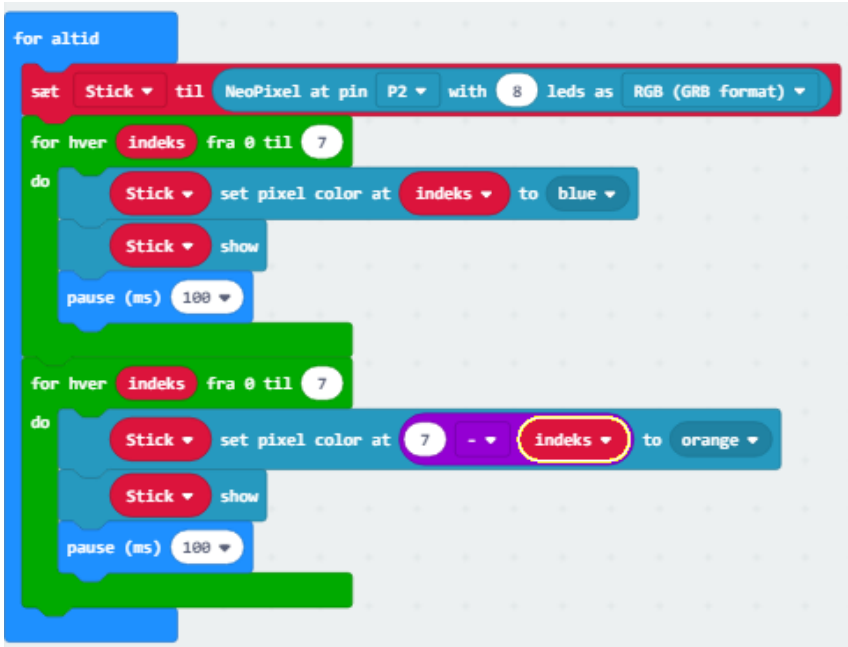
Du kan prøve at lægge koden ned på Micro:bit og se den live på din Fire Stick Gizmo.

Hvis du har lyst, kan du prøve at vende den nye løkke, så lyset løber den anden vej.

Her skal du bruge en klods fra MATEMATIK.

Sæt den ind i stedet for ”index” i ”set color at..”
Skriv tallet 7 i det første felt med 0 og sæt variablen ”index” ind ved det andet felt med 0.

Nu ser hele koden sådan ud:



Læg koden over på din Micro:bit og se hvordan lyset nu løber begge veje.

Playground og Crumbs.

Har du Crumbs liggende, kan du også bruge dem sammen med Playground og Micro:bit.

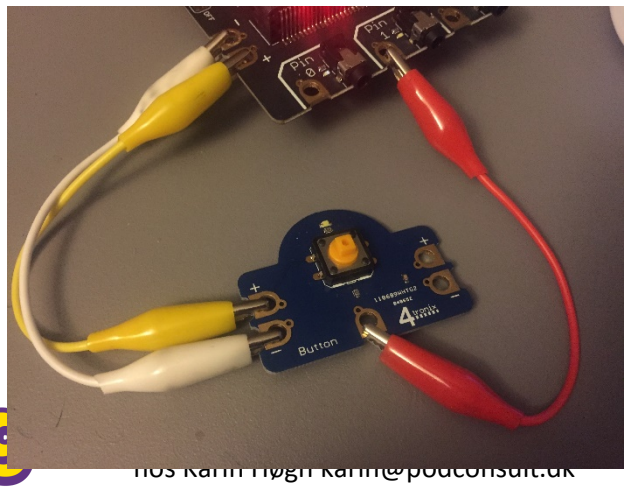
Nogle Crumbs skal bruge lidt mere strøm end Micro:bit kan give, men med Playground har jo sin egen strømforsyning på ryggen.

I det følgende får du en række eksempler på, hvordan du kan forbinde de ældre Crumbs. Du får også et par eksempler på at kode dem med Makecode og Micro:bit. Ofte kan du bruge samme kode, som dem du har set/lavet til Gizmos.

Forbind Crumbs til Playground.

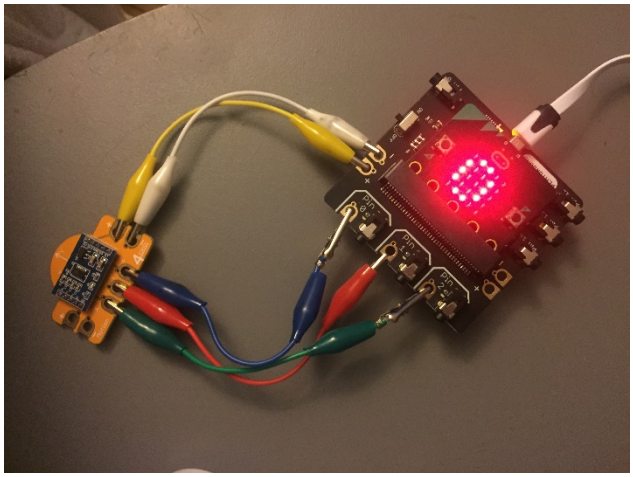
Til de fleste Crumbs skal du bruge 3 krokodillenæb. I enkelte tilfælde som f.eks. Trafic og Motion skal du bruge 5 krokodillenæb. Gizmos blev netop udviklet for at gøre det langt mere enkelt at overskue forbindelser mellem Playground og eksterne komponenter.

Eksempel med 3 krokodillenæb:



Forbindelserne fra plus (gul ledning) og minus (hvid ledning) på Playground til Crumb (her Button) giver næsten sig selv. Men der skal også et krokodillenæb fra bunden af din Crumb (rød ledning) til den pin du bruger på Playground (her Pin1).

Eksempel med 5 krokodillenæb:



Plus og minus forbindes som ovenfor. Motion og Trafic har brug for at forbinde til 3 pins samtidig. (rød, blå, grøn ledning)

Farver på Crumbs matcher farverne på Gizmos, så du kan gå ud fra at blå Crumbs er digitale input, gule Crumbs er analoge input, røde Crumbs er analoge output, sorte Crumbs til digitale output. Sparkles, der er pendanten til Flame Gizmo, er i Crumbs-serien sort.

Der findes flere Crumbs som endnu ikke er lavet som Gizmos, som f.eks. proximity, hall effect (magnet) og motion. Der findes flere Crumbs end vist nedenfor. Så længe Crumbs produceres kan de skaffes hjem hos www.itogleg.dk

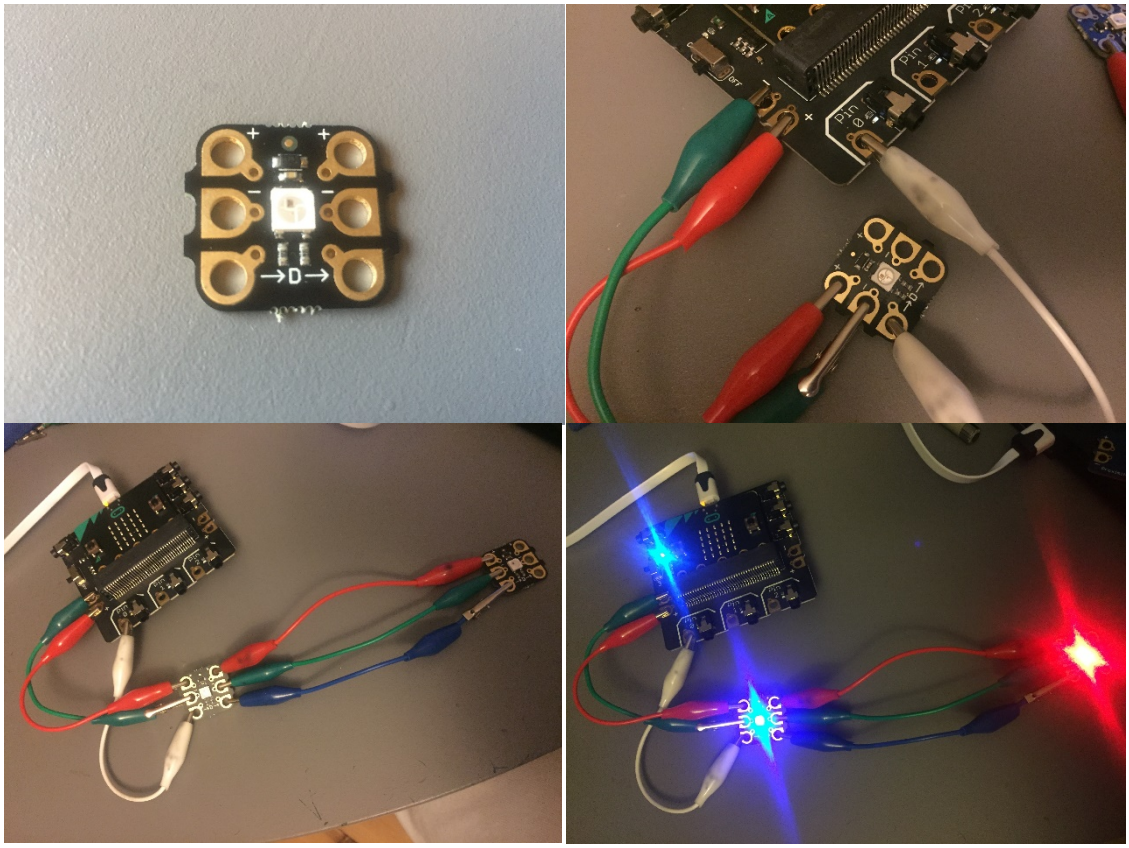
Eksempler på leg med Crumbs til Playground og Micro:bit

Sparkles

Forbind Sparkles

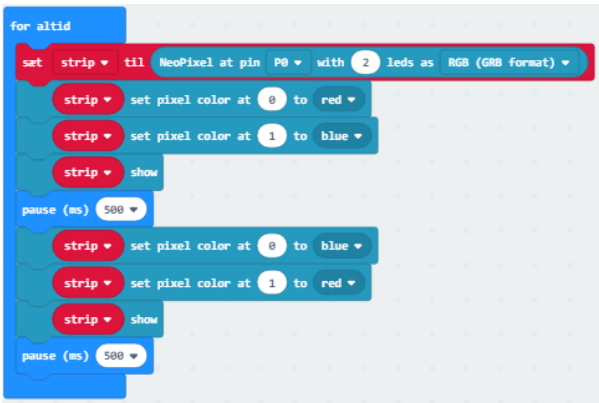
Du skal bruge 3 krokodillenæb for hver Sparkle du vil sætte ind i dit kredsløb. Se billederne nedenfor, hvor Sparkle nummer 1 er forbundet med Pin0 på Playground.
Sparkles skal ligesom Fire Stick og Flame Gizmos vende rigtigt i forhold til input fra Playground.

Øverst: På Sparkle finder du en pil, der angiver, hvilken retning din Sparkle skal forbindes. Strømmen (plus og minus) kan løbe begge veje, men data fra Pins 8her Pin0) kan kun løbe samme vej som pilen.
Nederst: Eksempel på flere Sparkles i en serie. Begge Sparkles skal vende i samme retning.



Kod Sparkles

Du kan kode Sparkles med de samme koder som du bruger til Fire Stick og Flame Gizmos. Her får du igen eksemplet fra tidligere til Flame Gizmo. Se evt. beskrivelsen der for at bygge koden op.

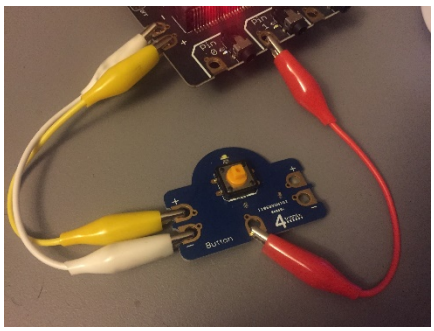


Flere eksempler på sorte Crumbs

Buzzer Crumb afgiver en lille lyd, når den for et digitalt output fra Micro:bit. Du kan bruge Buzzer crumb med både MUSIK (kun på Pin0) og med PINS (digital) på Pin0 og øvrige pins på Playground

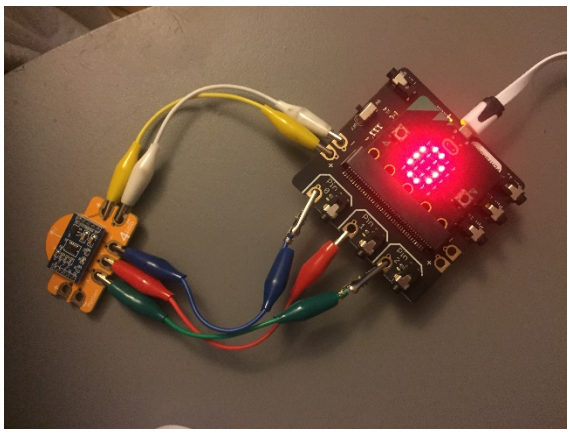
Trafic Crumb har 3 LED grøn, gul og rød monteret til at eksperimentere med skift af trafiklys.

Forbind Buzzer Crumb
- forbindes ligesom blå Crumbs i næste afsnit med 3 krokodillenæb.

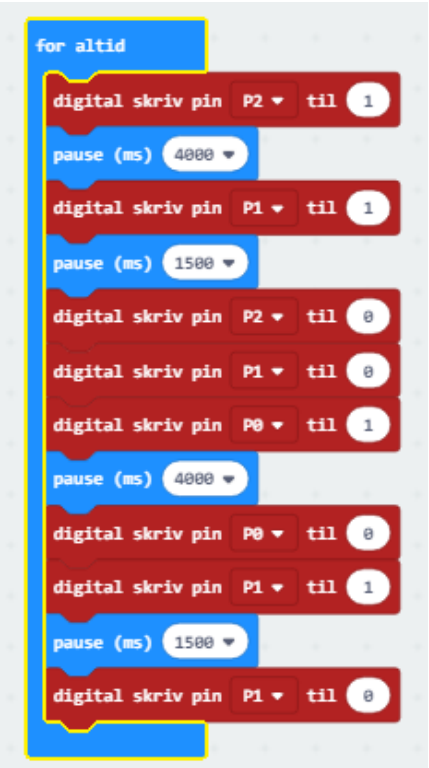


En kode til Buzzer Crumb (3 krokodillenæb) kunne se sådan ud:

Forbind Trafic Crumb
- forbindes med 5 krokodillenæb ligesom gul Motion Crumb på billedet nedenfor.



En kode til Trafic Crumb (5 krokodillenæb) kunne se sådan ud.



BLÅ Crumbs

Disse tre eksempler på blå Crumbs kan bruges med samme kode.

Proximity er en afstands-sensor, der registrerer når en hånd kommer tæt på sensoren (2-3 cm).

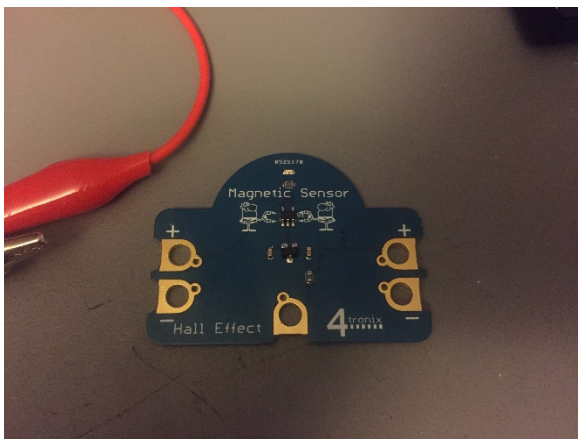
Hall Effect er en magnetisk sensor, der registrerer når du fører en magnet tæt på sensoren.

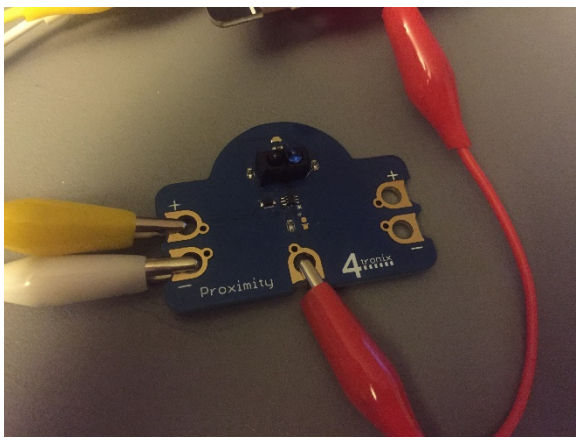
Button er en trykknap, der registrerer om knappen er trykket ned eller ikke.

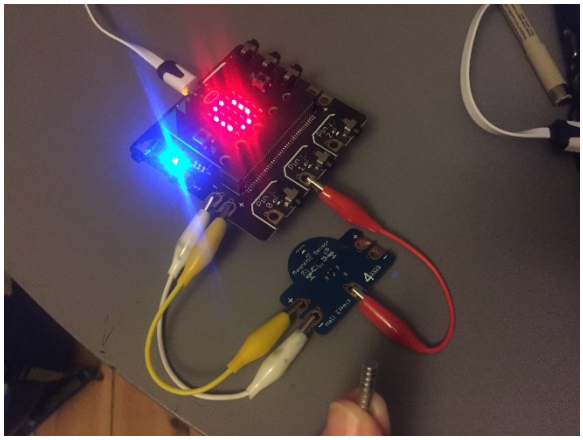
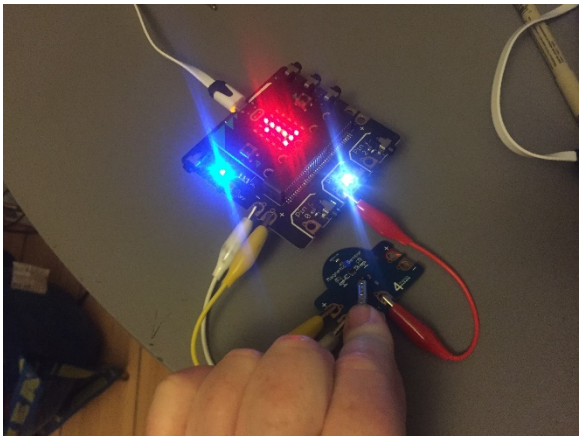
Forbind de blå Crumbs

Du skal bruge 3 krokodillenæb til at forbinde en af de tre Crumbs. Se billederne nedenfor, hvor Crumbs forbindes med Pin1 på Playground.

Øverst: Eksempel på forbindelser fra Playground til Crumbs
Nederst: Eksempel på brug af blå Crumbs





Kod de blå Crumbs

Eksempel på en kode, der kan anvendes af alle disse tre eksempler på sensorer.

Gule Crumbs

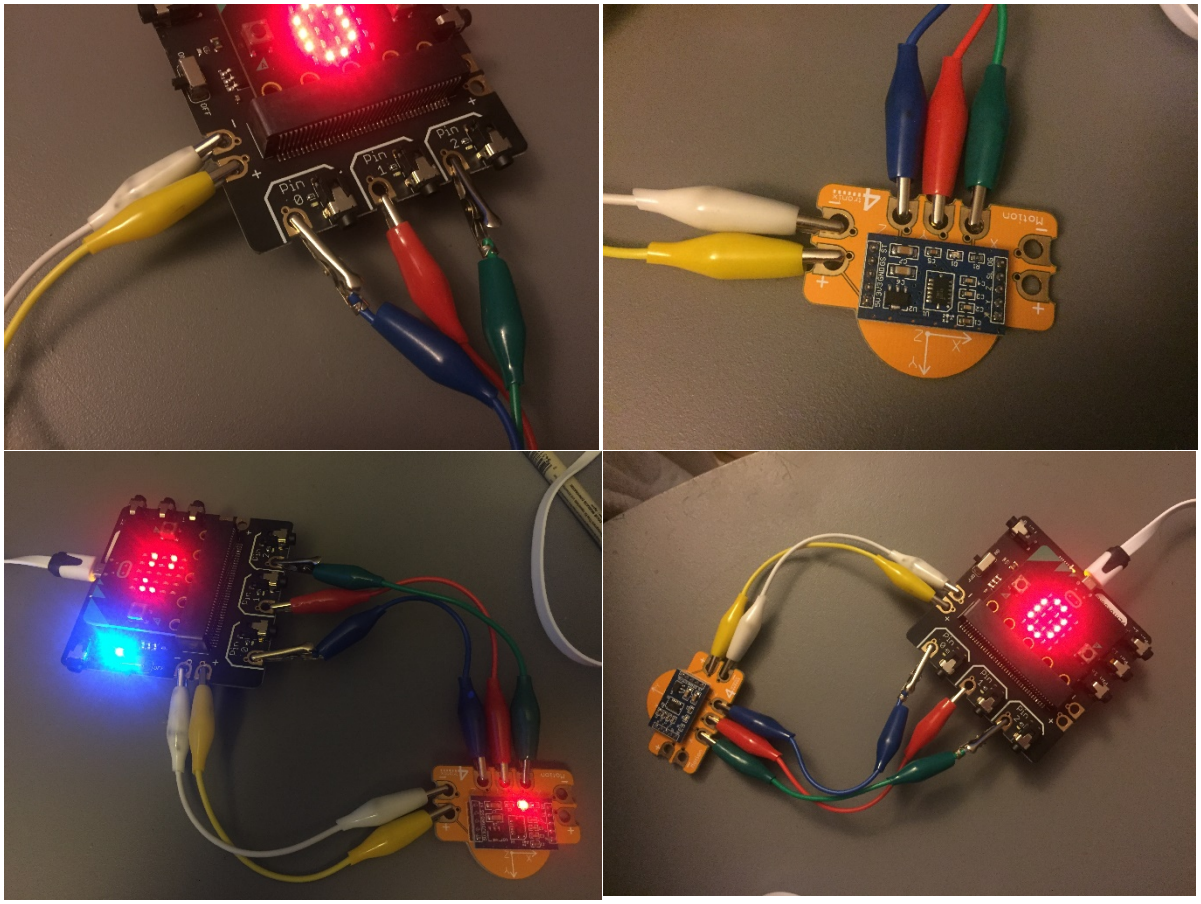
Eksempler på gule Crumbs:

Motion Crumb måler acceleration i tre retninger x, y, z.

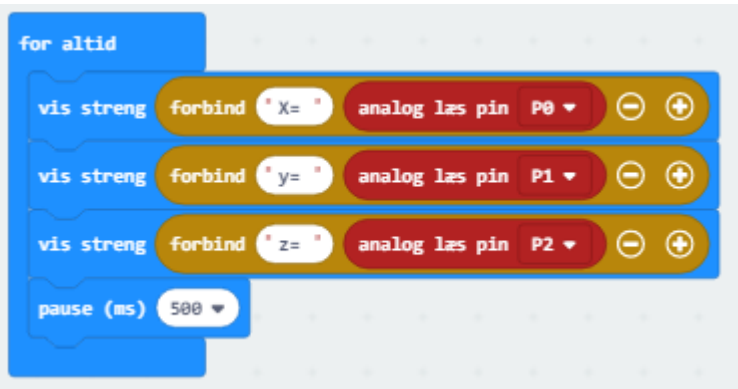
Dial Crumb er et potentiometer, hvor du med en drejeknap kan skrue op og ned for strømmen.

Forbind Motion Crumb med 5 krokodillenæb. Se billeder nedenfor:

Hver akse skal bruge en pin til at give input til Micro:bit. Når Motion Crumbs bevæges, ændres tallene i displayet.



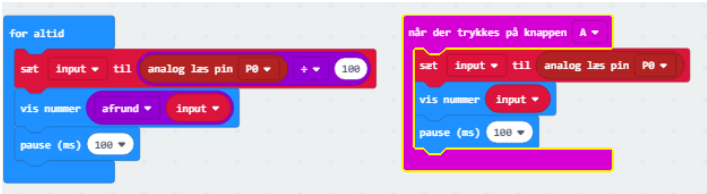
Kod Motion Crumb
-En kode til Motion Crumb kan se sådan ud:



De karrygule ”forbind..:” klodser finder du inde i TEXT biblioteket.

Forbind Dial Crumb med 3 krokodillenæb ligesom i eksemplerne med Blå Crumbs længere oppe i dokumentet.

En kode til Dial Crumb kunne se sådan ud (ligesom Dial Gizmo):

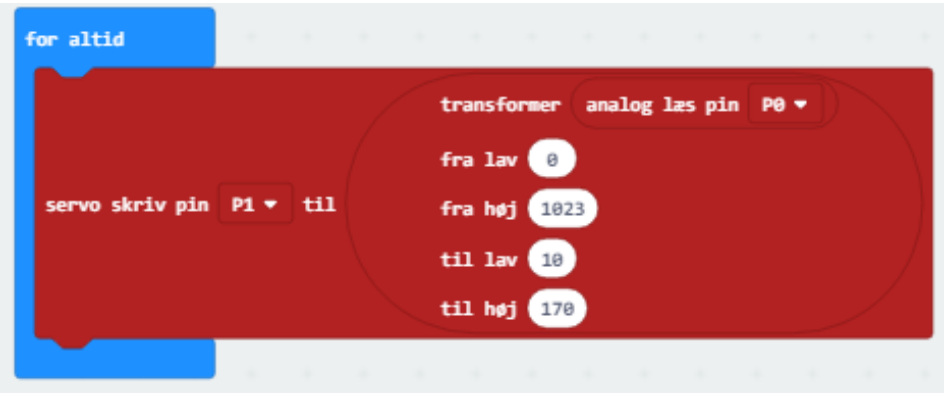


Røde Crumbs

Ligesom ved Gizmos kan du ikke køre motor Crumbs med Playground, men servo-motoren forbindes ligesom de øvrige Crumbs, der forbindes med 3 krokodillenæb

Kod Servo Crumb

Koden til en Servo Crumb kunne se ud som på næste side:



Se under Servo Gizmo her i guiden, hvis du vil have forklaret koden yderligere.

Eksperimentér!

Når du har leget lidt med de forskellige Gizmos og/eller Crumbs og lært dem at kende, kan du eksperimentere videre med at kombinere dem og få mere ud af sættene.

Input fra gule og blå Gizmos og Crumbs kan give Outputs til de røde og sorte – og hvide. Det er nu det kan blive rigtig sjovt, hvis du leger videre.

Eksperimentér også med at bygge dem ind i fysiske projekter med forskellige materialer og formål.

Denne guide er produceret og gennemprøvet i praksis af Sophie Hovdekorp, Kodesmart.dk i samarbejde med Micro-bit.dk.
Du er velkommen til at printe den ud, distribuere den, remixe den og bygge videre på arbejdet, så længe du krediterer os for det oprindelige værk.
For eksempel ved at skrive: Tak til Sophie Hovdekorp og Micro-bit.dk for materialet.

BBC micro:bit og tilbehør sælges
hos Karin Høgh karin@podconsult.dk