

MICRO:BIT

GRUNDKURS

MICRO:BIT

Utlåningslåda 1 - Grundläggande

Introduktion

Att jobba med programmering är spännande men kan kännas lite skrämmande till en början. Vi har därför tagit fram den här lådan för dig som lärare. Tanken är att på ett enkelt sätt, utan krav på förkunskaper, få dig att komma igång med programmering.

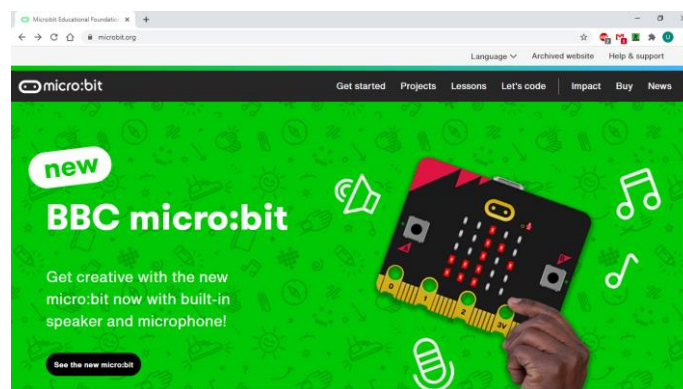
I lådan finns allt som behövs för att genomföra fem olika lektioner med eleverna. Du kanske känner dig lite osäker efter att ha öppnat lådan och tittat igenom instruktionerna? I så fall ska du veta att du gärna får kontakta oss på Upptech med frågor etc. så hjälper vi till.

Dator eller iPad?

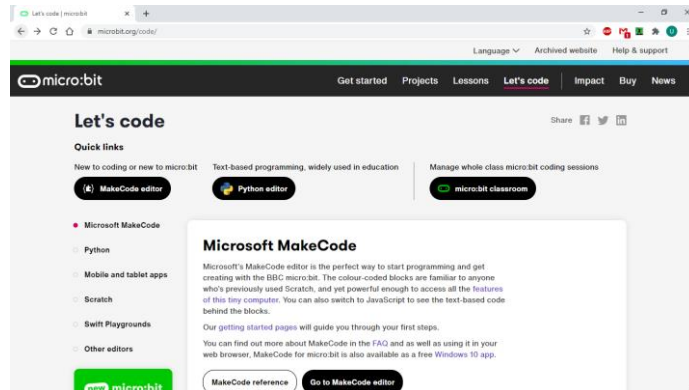
Micro:bit går bra att programmera med både dator och iPad. Det skiljer sig dock lite hur man kopplar upp sig mot micro:biten. Nedan följer instruktioner om hur man gör med båda.

Dator

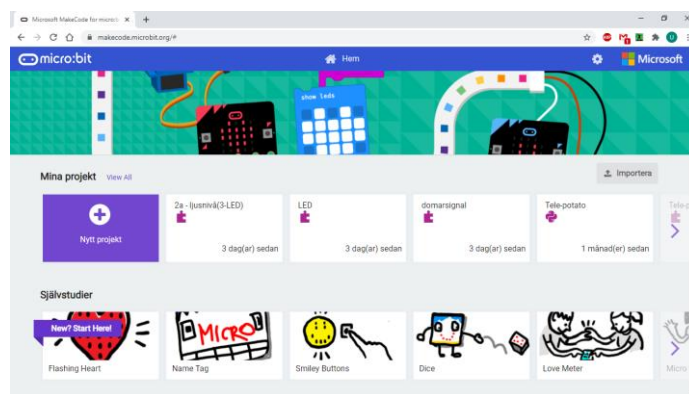
Gå in på <http://microbit.org/>



Koppla in din micro:bit med USB-kabeln och gå in på <http://microbit.org/code/>

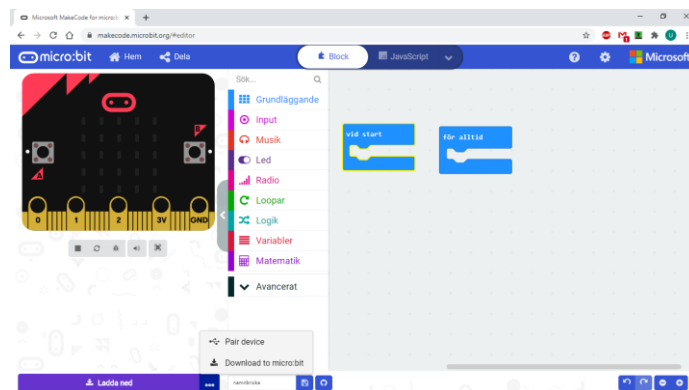


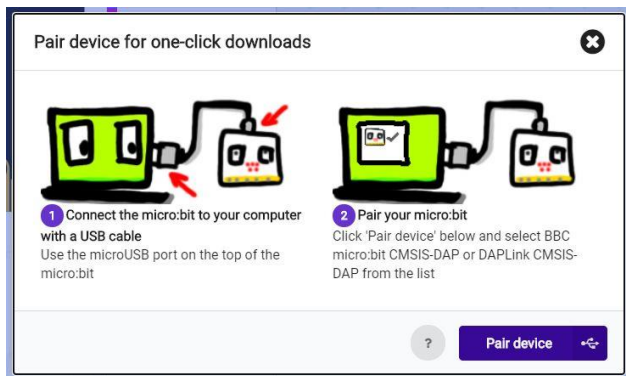
och under "New to coding or new to micro:bit", tryck på knappen märkt "MakeCode editor".



Om inte svenska är valt kan du ändra det genom att gå på kugghjulet till vänster om Microsoftloggan uppe till höger. Välj första alternativet "Language" och sedan svenska (vi rekommenderar dock att man använder engelska då det är vanligare inom programmering). Därefter välj "Nytt projekt".

Tryck sedan på de tre punkterna nere till vänster vid knappen "Ladda ned". Välj "Pair device".





Tryck "Pair device".

makecode.microbit.org vill ansluta



Välj micro:bit i listan och tryck "Anslut". Nu behöver du bara trycka den stora blå knappen "Ladda ned" för att föra över din programmering. Microbit:en har kopplats ihop med datorn!

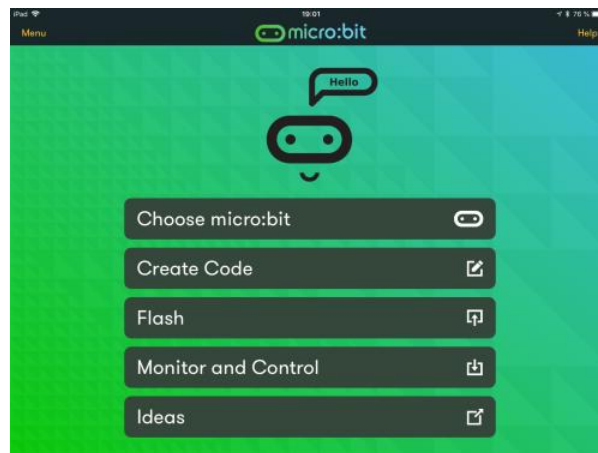
iPad

Om du använder iPad eller iPhone finns det en app som gör att du kan ansluta till en micro:bit. Anslutningen sker via bluetooth (trådlöst) och låter dig programmera din micro:bit. Appen ser ut så här i JKPG Appar:

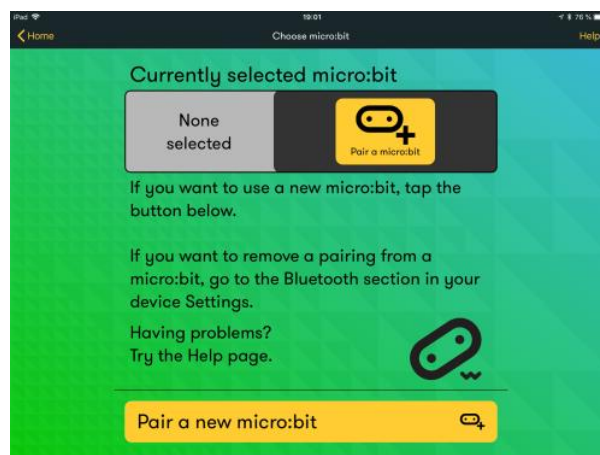


(eventuellt står det ingen text)

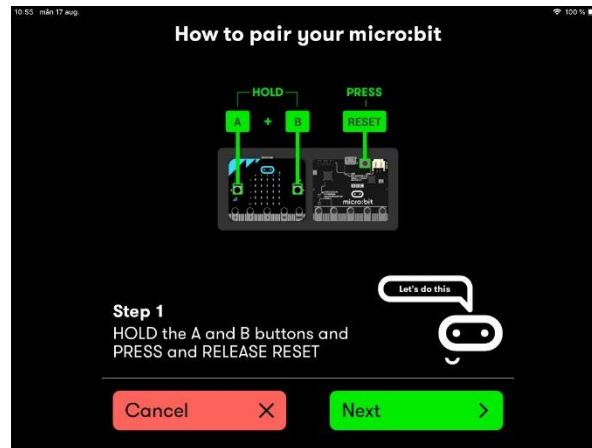
Första gången du startar appen väljer du "Choose micro:bit".



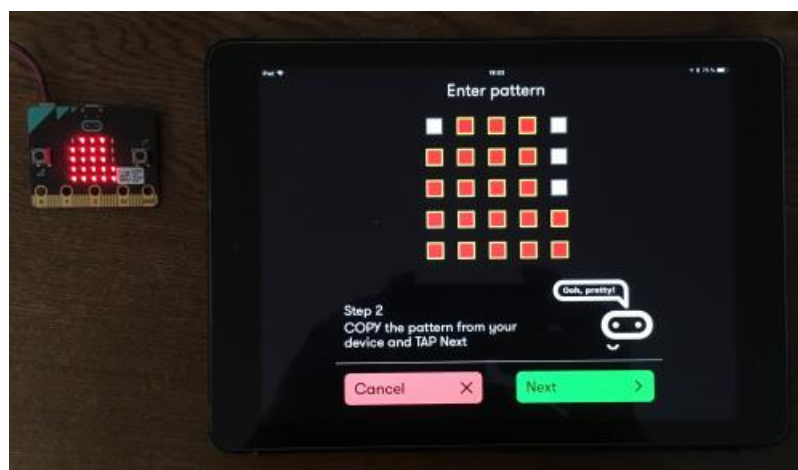
På sidan som öppnas väljer du "Pair a new micro:bit".



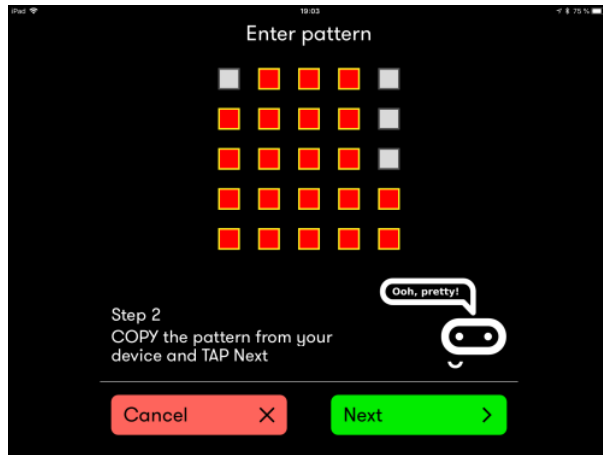
Nu kommer ett steg som kräver lite fingerfärdighet. Håll nere knapparna A och B på din micro:bit. Medan du håller dem nere trycker du ner och släpper upp resetknappen på baksidan av din micro:bit.



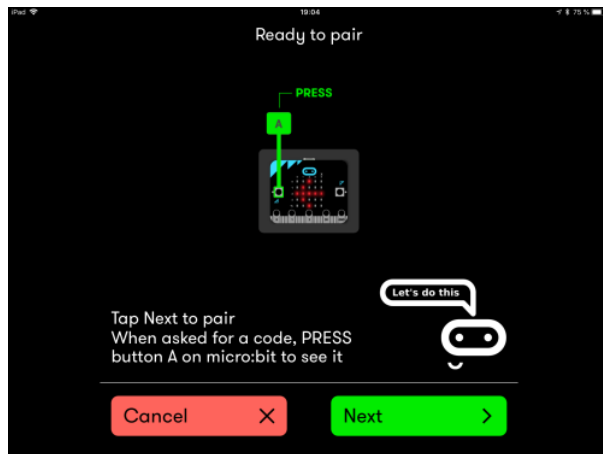
Om du gjort rätt visar sig fem olika staplar på displayen mellan A och B-knapparna på din micro:bit.



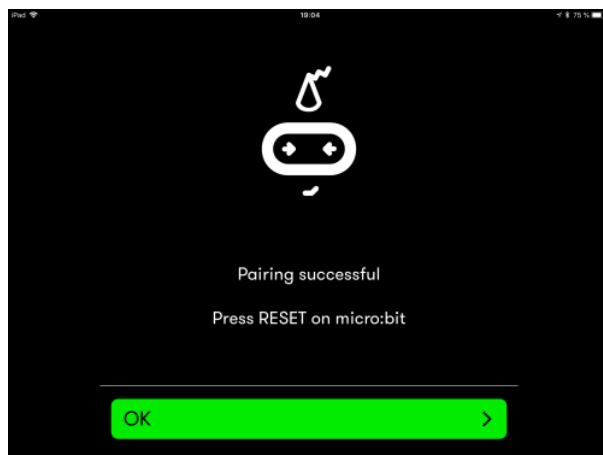
Ställ in staplarna som visas i appen på samma sätt som de visas på din micro:bit och tryck på "Next".



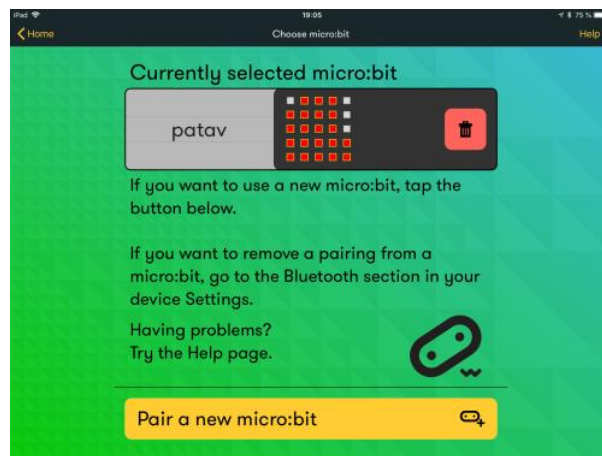
På nästa sida står det att du ska trycka på knapp A när appen frågar efter en kod. Det behöver du inte göra. Tryck bara på "Next".



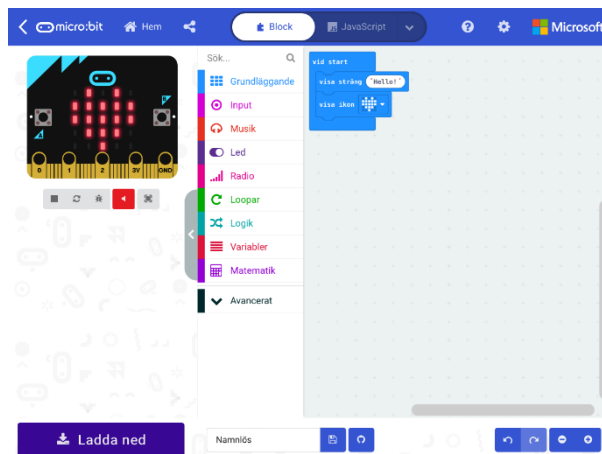
Du kan få en fråga om "Bluetooth Pairing Request". Välj i så fall "Pair". Om allt gått bra får du meddelandet "Pairing successful". Tryck på OK.



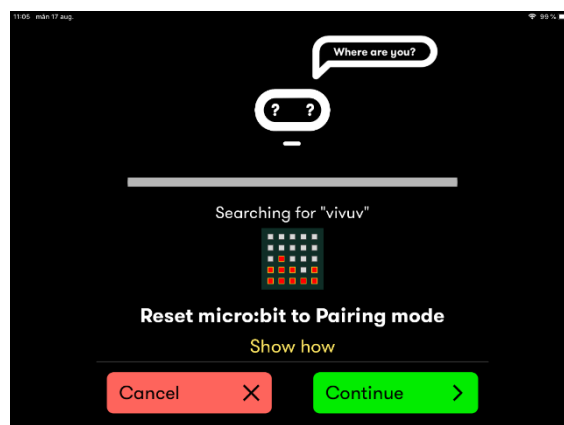
Nu visas en sida där det står "Currently selected micro:bit" och under det visas mönstret som du tryckte in. Nu kan du klicka på "Home" högst uppe till vänster och fortsätta med programmering.



Efter du gjort din programmering vill du överföra den till micro:biten.

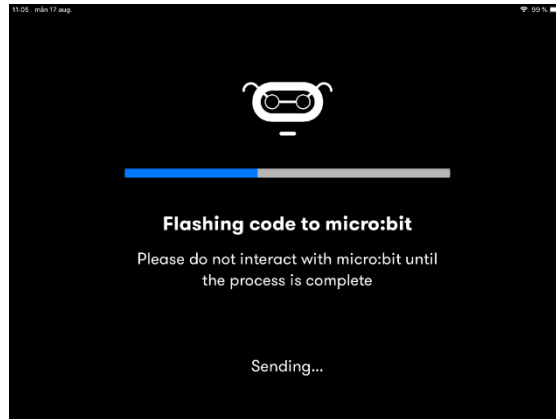


Tryck "Ladda ned" och du kommer till nästa skärm (notera att det är en annan micro:bit nu, kallad "vivuv" isf. "patav").

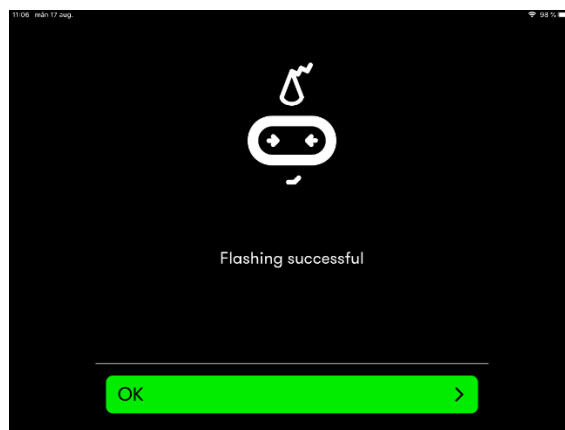


Programmeringen kan inte överföras för att iPaden inte hittar micro:biten. För att lösa detta, tryck på A + B + Reset samtidigt på micro:biten. Släpp därefter Reset och håll kvar tills micro:biten visar bluetoothikonen på skärmen.

Därefter tryck på "Continue" på iPaden. Programmeringen överförs.



När överföringen är klar ser du denna skärm och programmeringen fungerar på din iPad:



Microbit:en har kopplats ihop med iPaden!

Notera: man kan med fördel trycka A + B + Reset på micro:biten precis innan man trycker på "Ladda ned" på iPaden. Då överförs programmeringen direkt.

MICRO:BIT

Utlåningslåda 1 - Grundläggande

Introduktion

Att jobba med programmering är spännande men kan kännas lite skrämmande till en början. Vi har därför tagit fram den här lådan för dig som lärare. Tanken är att på ett enkelt sätt, utan krav på förkunskaper, få dig att komma igång med programmering.

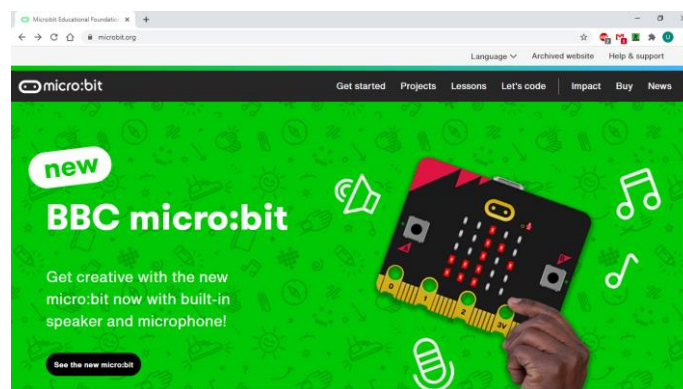
I lådan finns allt som behövs för att genomföra fem olika lektioner med eleverna. Du kanske känner dig lite osäker efter att ha öppnat lådan och tittat igenom instruktionerna? I så fall ska du veta att du gärna får kontakta oss på Upptech med frågor etc. så hjälper vi till.

Dator eller iPad?

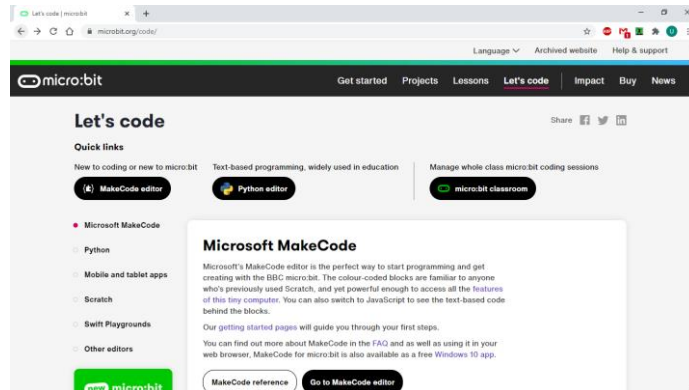
Micro:bit går bra att programmera med både dator och iPad. Det skiljer sig dock lite hur man kopplar upp sig mot micro:biten. Nedan följer instruktioner om hur man gör med båda.

Dator

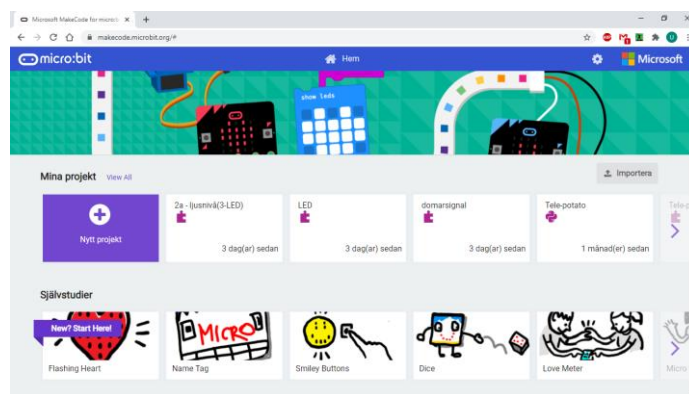
Gå in på <http://microbit.org/>



Koppla in din micro:bit med USB-kabeln och gå in på <http://microbit.org/code/>

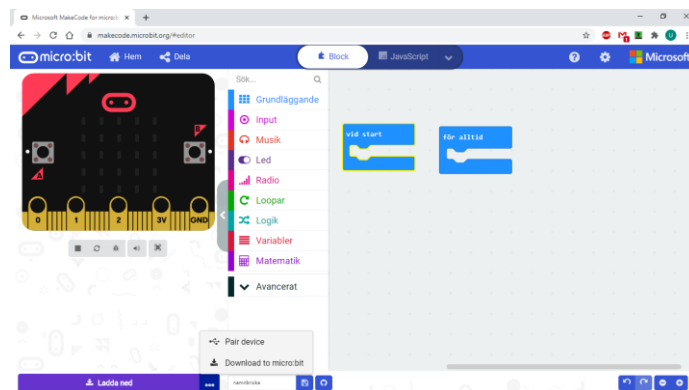


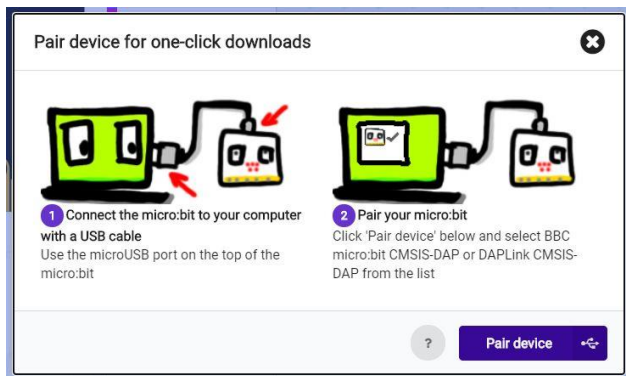
och under "New to coding or new to micro:bit", tryck på knappen märkt "MakeCode editor".



Om inte svenska är valt kan du ändra det genom att gå på kugghjulet till vänster om Microsoftloggan uppe till höger. Välj första alternativet "Language" och sedan svenska (vi rekommenderar dock att man använder engelska då det är vanligare inom programmering). Därefter välj "Nytt projekt".

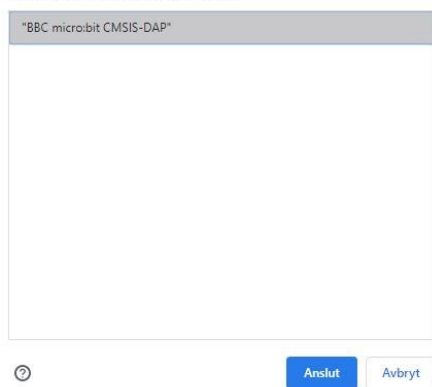
Tryck sedan på de tre punkterna nere till vänster vid knappen "Ladda ned". Välj "Pair device".





Tryck "Pair device".

makecode.microbit.org vill ansluta



Välj micro:bit i listan och tryck "Anslut". Nu behöver du bara trycka den stora blå knappen "Ladda ned" för att föra över din programmering. Microbit:en har kopplats ihop med datorn!

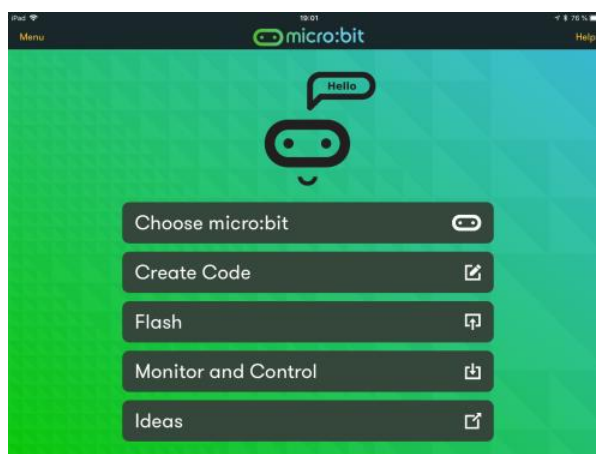
iPad

Om du använder iPad eller iPhone finns det en app som gör att du kan ansluta till en micro:bit. Anslutningen sker via bluetooth (trådlöst) och låter dig programmera din micro:bit. Appen ser ut så här i JKPG Appar:

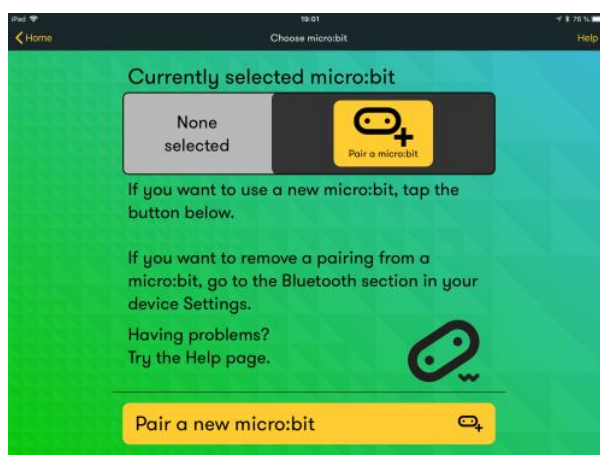


(eventuellt står det ingen text)

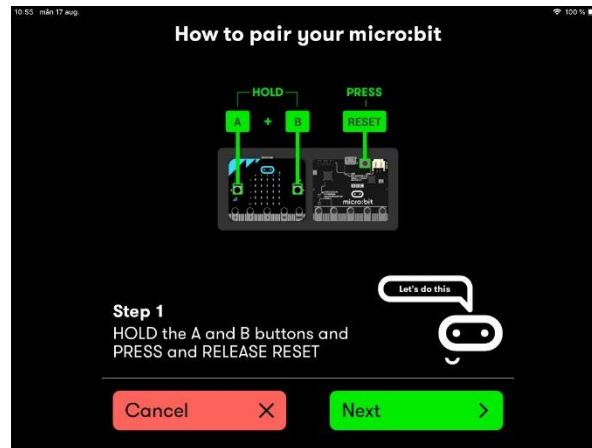
Första gången du startar appen väljer du "Choose micro:bit".



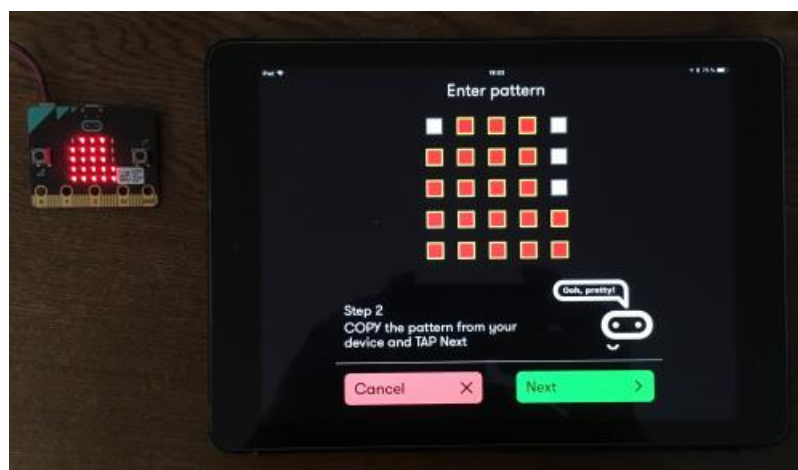
På sidan som öppnas väljer du "Pair a new micro:bit".



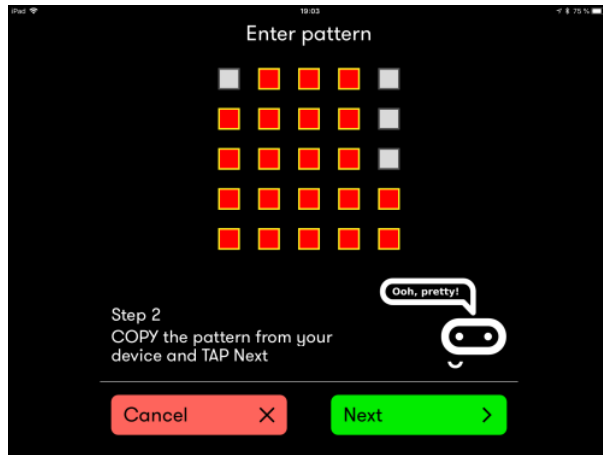
Nu kommer ett steg som kräver lite fingerfärdighet. Håll nere knapparna A och B på din micro:bit. Medan du håller dem nere trycker du ner och släpper upp resetknappen på baksidan av din micro:bit.



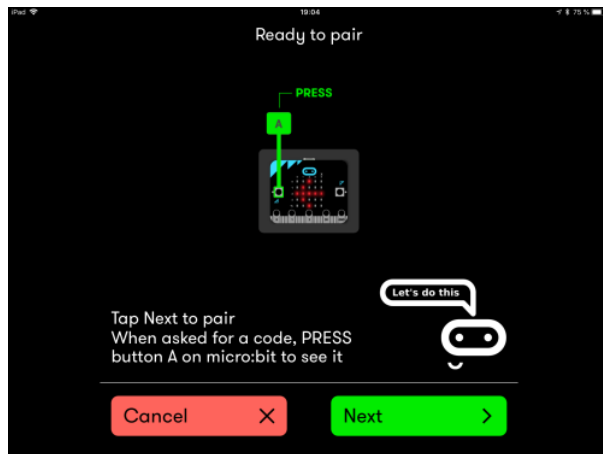
Om du gjort rätt visar sig fem olika staplar på displayen mellan A och B-knapparna på din micro:bit.



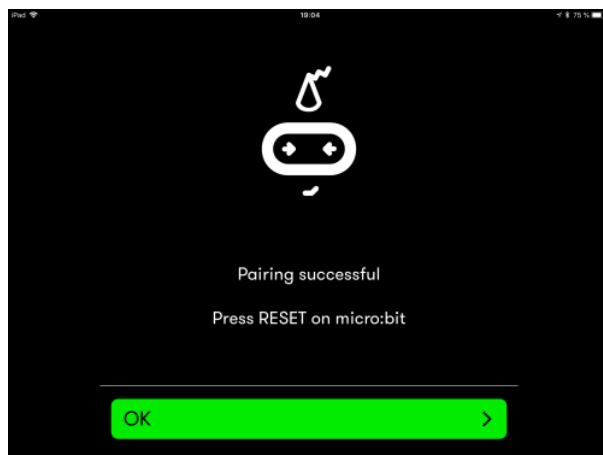
Ställ in staplarna som visas i appen på samma sätt som de visas på din micro:bit och tryck på "Next".



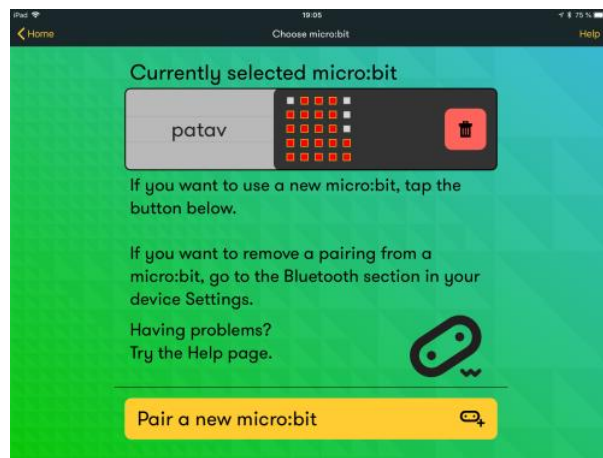
På nästa sida står det att du ska trycka på knapp A när appen frågar efter en kod. Det behöver du inte göra. Tryck bara på "Next".



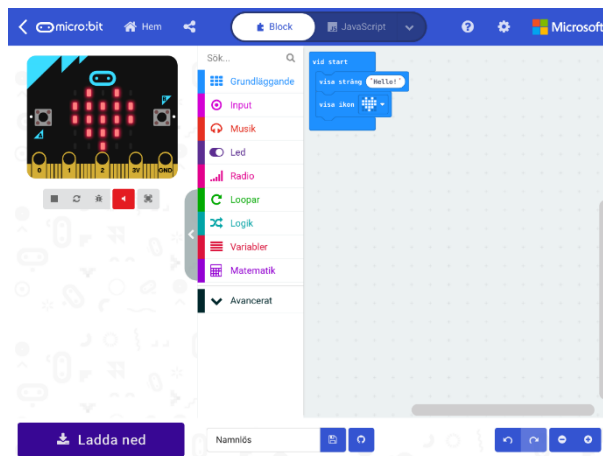
Du kan få en fråga om "Bluetooth Pairing Request". Välj i så fall "Pair". Om allt gått bra får du meddelandet "Pairing successful". Tryck på OK.



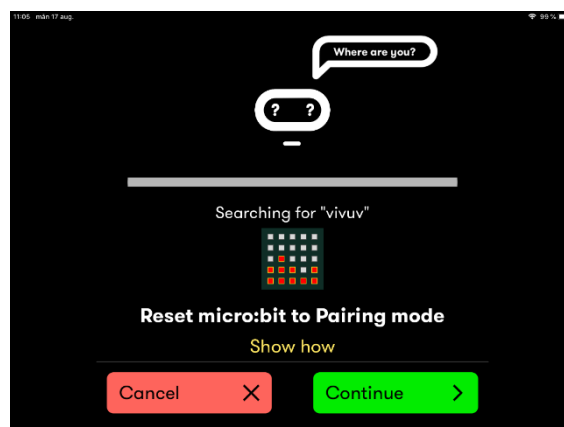
Nu visas en sida där det står "Currently selected micro:bit" och under det visas mönstret som du tryckte in. Nu kan du klicka på "Home" högst uppe till vänster och fortsätta med programmering.



Efter du gjort din programmering vill du överföra den till micro:biten.

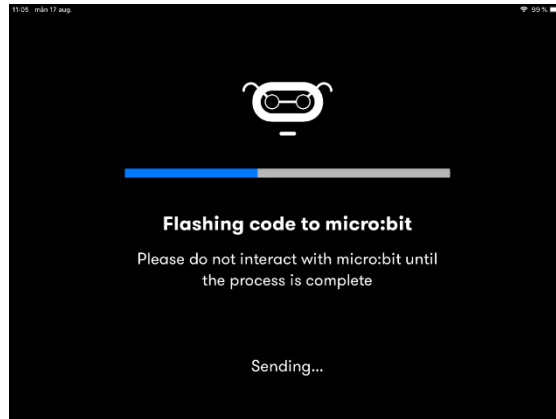


Tryck "Ladda ned" och du kommer till nästa skärm (notera att det är en annan micro:bit nu, kallad "vivuv" isf. "patav").

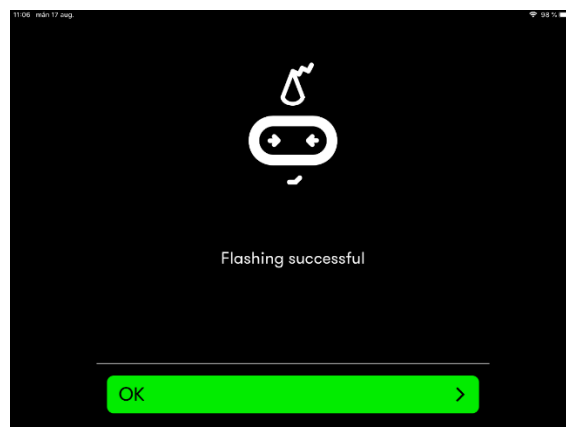


Programmeringen kan inte överföras för att iPaden inte hittar micro:biten. För att lösa detta, tryck på A + B + Reset samtidigt på micro:biten. Släpp därefter Reset och håll kvar tills micro:biten visar bluetoothikonen på skärmen.

Därefter tryck på "Continue" på iPaden. Programmeringen överförs.



När överföringen är klar ser du denna skärm och programmeringen fungerar på din iPad:



Microbit:en har kopplats ihop med iPaden!

Notera: man kan med fördel trycka A + B + Reset på micro:biten precis innan man trycker på "Ladda ned" på iPaden. Då överförs programmeringen direkt.

MICRO:BIT

Utlåningslåda 2 - Fortsättning

Uppgifter

Denna låda innehåller extrasaker som man kan koppla till en micro:bit. Det som finns i lådan är:

1. 15 x 6 st krokodilkablar
2. 15 x 5 st stora lysdioder (LED). Grön, röd, gul, blå och vit.
3. 15 x 1 st summer

Till varje micro:bit ska man alltså ha 6 st krokodilkablar, 4 st LED (en vit, en grön, en gul och en röd) samt en summer.

Vi har valt att dela upp innehållet i fem lektioner enligt följande:

1. Summer eller LED.
2. Summer och LED.
3. Ljusmätare med tre nivåer.
4. Ljusmätare med tre LED.
5. Vandrande LEDs (disco).

Nedan följer mer utförlig beskrivning av de olika lektionerna.

Lektion 1: Summer eller LED.

Uppdrag: Programmera så att du kan styra en LED av och på med knappar.

När man vill koppla in annan utrustning är krokodilkablar ett enkelt sätt att börja med. En krokodilkabel ser ut så här:



Styrning av en summer/LED

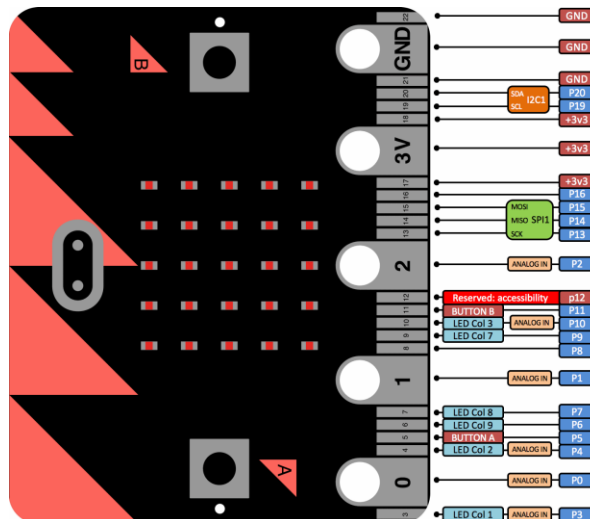
En summer är en elektrisk komponent som skapar ett enkelt ljud när en drivspänning ansluts. Vi ska ansluta den till en av micro:bitens s.k. "pins". Dessa finns längst ner på micro:biten.



Det finns tre lättåtkomliga pins, märkta med "0", "1" och "2". Vi har även två till; "3V" och "GND".

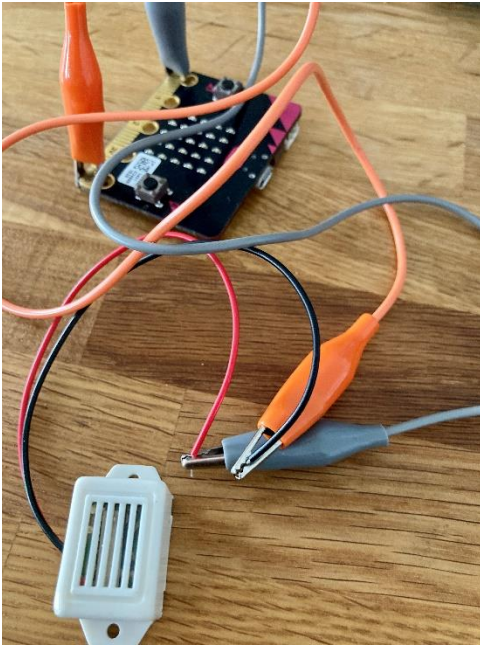
- 0, 1 och 2 Pins som kan skicka ut information.
- 3V Pin som kan skicka ut 3V spänning till externa saker som behöver extra mycket spänning.
- GND Pin för återkoppling och kretskoppling. GND står för GROUND (jord på svenska).

Det finns fler pins på en micro:bit. Alla små fält är en pin men för att kunna använda sig av dem bör man koppla micro:biten till någon typ av moderkort. Totalt finns det 25 stycken pins:



För att koppla upp en summer behöver vi koppla med krokodilkablar enligt följande:

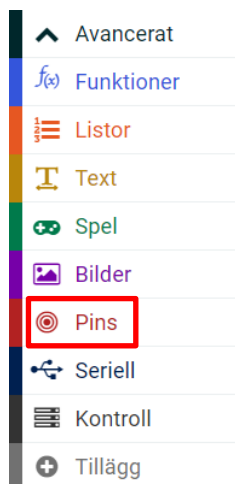
- Koppla ut från en av micro:bitens pin 0, 1 eller 2 till den röda kabeln på summern.
- Koppla från den svarta kabeln på summern till pin GND på micro:biten.



Notera att det kan vara krångligt att få krokodilkontakten så att den enbart täcker pin 0 respektive pin GND. Vi brukar fästa kontaktens ena käft i hålet så håller den sig på rätt plats.

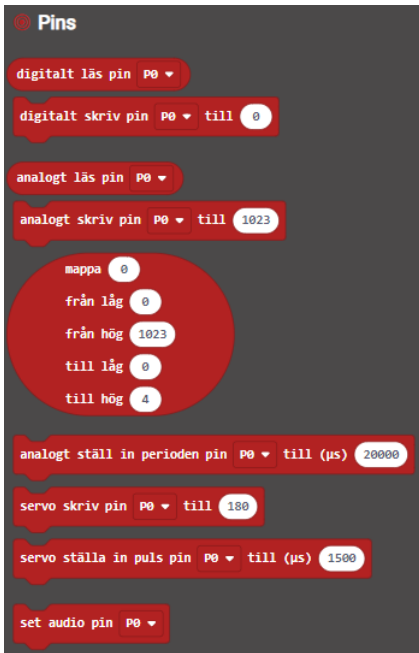
Programmering:

När man vill programmera pins finns det en speciell blockrubrik som hanterar just dessa.



Om man trycker på "Avancerat" så öppnas ett antal fler blockrubriker upp (se till vänster).

Under "Pins" finns de block som vi ska använda:

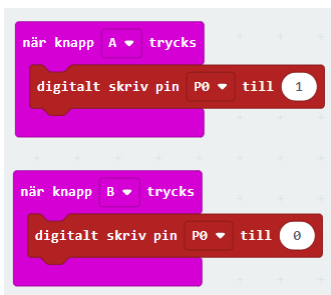


Det vi vill göra är att styra summern med hjälp av knapparna på micro:biten. A ska slå på summern och B ska stänga av summern.

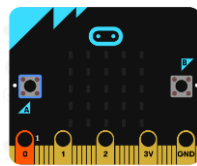
Digital styrning:

För att göra det enkelt kan vi använda s.k. digital styrning vilket innebär två lägen "på" och "av". Det innebär att när värdet är 1 är summern på och när värdet är 0 så är summern av (binär styrning).

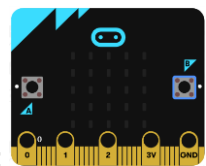
Programmeringen blir då så här:



Knapp A tryckt:



Knapp B tryckt:



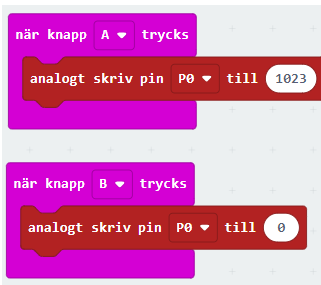
Bilderna visar hur det ser ut i simulatören när vi provar programmeringen. Efter vi tryckt på knapp A så lyser pin 0 orange och det står en 1:a vid den. Trycker vi därefter på knapp B så slocknar pin 0 och det står en 0:a vid den. Uti från det kan vi konstatera att programmeringen fungerar.

Ladda över programmeringen till micro:biten (som är kopplad till summern enligt sida 2) och prova!

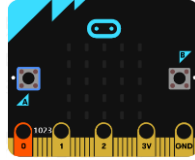
Analog styrning:

När man styr analogt finns det fler än två nivåer att välja på. Närmare bestämt 1024 stycken. Istället för ett "digitalt skriv pin"-block väljer vi "analogt skriv pin"-block.

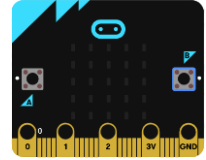
Programmeringen blir då så här:



Knapp A tryckt:



Knapp B tryckt:



Då det finns 1024 nivåer innebär det att spännet är 0 – 1023. Man kan alltså variera spänningen som skickas till summern genom att inte skicka max (1023) till summern.

Vilket värde för A kan du hitta som gör att summern vibrerar men inte låter?

Prova med en LED:

I lådan finns även lysdioder (LED eller Light Emitting Diode) i olika färger. De går att använda med samma koppling och programmering som vi gjort tidigare. Men var ska vi koppla krokodilkablarna till LED?



En lysdiod har två olika långa ben. Det långa benet kallas "Anod" och är positivt (+) medan det korta benet kallas "Katod" och är negativt (-). På summern hade vi sladdar som var färgmärkta med röd för positivt och svart för negativt.

Vi kopplar alltså från pin 0 till det långa benet och pin GND till det korta benet på LED.

Vad händer om du kopplar tvärtom (dvs. pin 0 till korta benet och pin GND till långa benet)?

Lyser alla LED lika starkt med samma programmering?

Lektion 2: Summer och LED.

Uppdrag: Programmera en nedräkning och därefter larmar summern.

Programmering:

Vi använder oss av en röd LED och en summer. Programmeringen behöver göra en nedräkning som visas som blink på LED och därefter larmar summern. Vi vill inte att programmeringen ska upprepas utan bara en gång. Vill vi starta om den kan vi göra det med reset-knappen på baksidan av micro:biten.

För att få LED att blinka till behöver vi tända den först, pausa en liten stund och därefter släcka den.


```
vid start
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
```

Detta gör att LED blinkar en gång. Vi vill att den ska blinka tre gånger så vi repeterar detta tre gånger. Antingen genom att lägga till likadana block eller genom att använda ett loop-block som vi sätter till tre repetitioner.

```
vid start
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
pausa (ms) 500
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
pausa (ms) 500
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
pausa (ms) 500
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
pausa (ms) 500
```

```
vid start
upprepa 3 gånger
gör
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
pausa (ms) 500
```

eller med loop-block:

Här ser man vilken skillnad det kan bli i längd på programmeringen om man väljer att upprepa med loop-blocket. På programmeringsspråk kallas det DRY (Don't Repeat Yourself) och är bra att tänka på för att få till en smidig programmering som är lätt att följa och förstå. Jag väljer att gå vidare med den versionen.

Efter blinkandet vill vi att summern ska larma. Summern blir vi tvungen att koppla till en annan pin än LED för att inte summern ska låta då LED blinkar.

```
vid start
upprepa 3 gånger
gör
digitalt skriv pin P0 till 1
pausa (ms) 500
digitalt skriv pin P0 till 0
pausa (ms) 500
digitalt skriv pin P1 till 1
visa ikon
pausa (ms) 1000
digitalt skriv pin P1 till 0
Rensa skärmen
```

Koppling: När vi ska koppla in två komponenter till en micro:bit så använder vi två olika pin för att koppla ut till komponenterna medan vi återkopplar till micro:biten via en och samma pin, GND.

I programmeringen finns det två stycken skärm-block; ett kryss och ett "rensa skärmen"-block. Vad händer om du lägger kryssset innan "digitalt skriv pin p1 till 1"-blocket? Fungerar det lika bra?

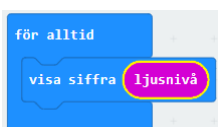
Lektion 3: Ljusbmätare med tre nivåer.

Uppdrag: Programmera så att en LED reagerar på ljuset i rummet. Om ljuset blir för mörkt ska LED tändas upp.

Programmering:

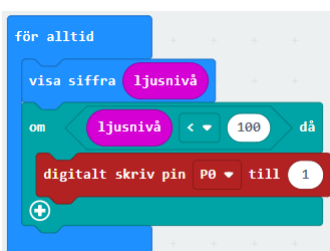
En ljusbmätare mäter ljusnivån och micro:biten har inbyggda ljussensorer i sin LED-matris på framsidan. I låda 1 fanns en uppgift där man gjorde en temperaturmätare och det går att applicera även på ljusnivå.

Det enklaste är att göra en programmering där ljusnivån automatiskt uppdateras hela tiden:




Micro:biten kommer då att visa på displayen hur ljust det är från 0 (mörkt) till 255 (ljust). Kan man bygga vidare på detta? Visst kan man det!

Vi kan införa ett villkor i vår programmering som bestämmer att om ljusnivån är för låg så tänds en lampa. En s.k. "nightlight". Utifrån det vi lärt oss i lektion ett vet vi hur man tänds en lampa så vi behöver nu ett villkor. Det vanligaste villkorsblocket är "if"-blocket ("om"-blocket på svenska). Det första blocket "visa siffra" låter vi ligga kvar då det kan ge oss ljusvärdet på micro:bitens skärm.



Om man provar denna programmering kommer man märka att LED tänds om det blir mörkt men om det därefter blir ljusare (över ljusvärde 100) kommer inte LED att släckas. Det beror på att vi enbart programmerat att LED ska tändas, inte att den ska släckas.

Tänk på att detta går att kontrollera i simulatoren innan man laddar över till micro:biten. Man kan justera en fiktiv ljusnivå i simulatoren via en cirkel upp till höger. 

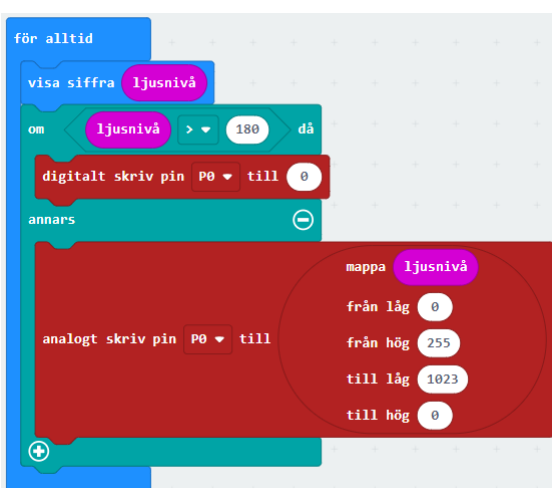


Nu har vi lagt in i programmeringen att om ljusvärdet går över 99 så släcks LED. Villkoret "annars" får man fram genom att trycka på +-tecknet.

Efter att ha kontrollerat att programmeringen fungerar i simulatorm så kan du ladda över programmet till micro:biten, koppla in LED (P0 till långt ben och GND till kort ben) och prova.

Fungerar programmeringen? Prova justera ljusnivågränsen så att den passar till rummet du är i.

Överkurs:



Här är en vidarebearbetning på programmeringen där LED släcks om ljusvärdet är över 180. Om det är under 180 så "mappas" värdet till utspänningen till LED. Det betyder att ju mörkare det blir i rummet (lägre ljusvärde) desto mer kommer LED att lysa.

I mappblocket väljs först "ljusnivå" att mappa. "från låg" och "från hög" hänvisar till micro:bitens inbyggda ljussensor och den mäter i 256 steg (0 – 255). Däremot skickas mappningen ut till en LED som (pga. analog styrning) har 1024 steg (0 – 1023). Att "till låg" sätts till det högsta (1023) och "till hög" till det lägsta (0) är för att vi vill att LED ska lysa när det är mörkt och slockna när det är ljust.

Här går det också att experimentera.

Vad händer om man kopplar in summern istället för en LED?

Lektion 4: Ljusbmätare med tre LED.

Uppdrag: Programmera flera LED så att de reagerar utifrån olika villkor (ljusnivåer). Rött när det är för mörkt, gult när det är mellannivå bra och grönt när det är ljust.

Repetition:

När man jobbar med villkor i programmering kan man använda t.ex. if-satser. Det har vi gjort under lektion 2 men nu ska vi bygga vidare på det. Först lite repetition.

If-satser fungerar på det sättet att man talar om ett villkor t.ex. "Om ljusnivån är 50 så ska det hända något speciellt". I vår programmering behöver vi då använda ett if-block som finns under "Logik".



Så här ser ett if-block ut (om-block på svenska). Det ligger i ett "för alltid"-block för if-blocket behöver ett start-block och kan inte ligga själv i programmeringen. Som blocket fungerar i den här situationen så behöver villkoret vara "sant" för att hjärtat ska visas.

Vad är det då som ska vara "sant"? Vi kan lägga i andra block i if-blocket för att kunna bestämma vårt villkor på ett bättre sätt och koppla det till vårt ljusvärde (ljusnivå).



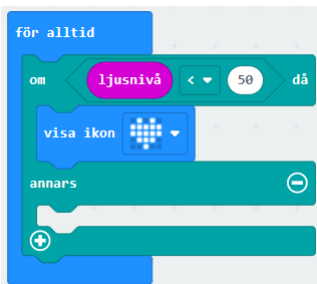
och vi lägger blocken på plats:



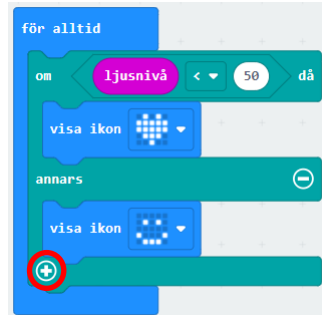
Nu har vi satt upp ett eget villkor: "Om ljusnivån är lägre än 50 ska hjärtat visas".

Om man nu provar detta i simulatören eller laddar över det till micro:biten kommer man upptäcka att hjärtat kommer att lysa upp när det blir mörkt men om det sedan blir ljust igen kommer hjärtat fortfarande att ligga kvar. Detta är för att vi inte har ett villkor för när hjärtat ska tas bort (för att ljusnivån blir högre).

För att lösa det inför vi villkoret "else" ("annars" på svenska). Det betyder att om inte if-villkoret är uppfyllt (ljusnivån är under 50) så talar vi om vad som ska ske. Genom att trycka på ++-knappen i if-blocket skapas automatiskt ett "else"-villkor (se röd markering ovan).



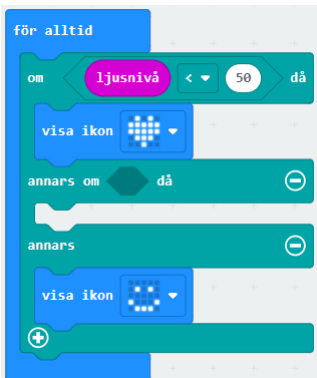
och vi kan tala om vad som ska ske:



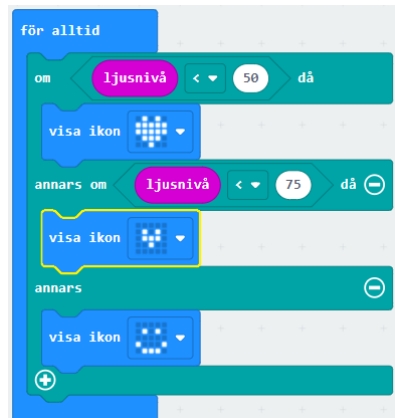
Nu fungerar programmeringen så att: om ljusnivån är under 50 (49 till 0) så kommer hjärtat att visas men om ljusnivån är 50–255 kommer glad gubbe visas. Eftersom programmeringen ligger i ett "för alltid"-block kommer programmeringen att anpassa sig så länge den körs.

Fortsättning:

Om man vill ha fler villkor än två går det att skapa genom att trycka än en gång på +-tecknet längst ner till vänster i if-blocket (se röd markering ovan):



och modifiera det nya villkoret:



Hur tror du den nya programmeringen uppför sig vid: ljusnivå 49, ljusnivå 74 och ljusnivå 99?

Programmering:

Vi ska använda tre olika LED så vi behöver ha tre villkor i vår if-sats.



Vilka exakta nivåer kan vi ändra senare (beroende på vilka ljusförutsättningar vi har i lokalen). Jag har satt villkor 1 till att vara under 60, villkor 2 till under 85 och villkor tre till under 255 (som är maxnivå).

Vi ska senare fysiskt koppla in en röd, en gul och en grön LED så vi behöver ha tre olika pins som styr respektive LED. Enklast är att välja 0, 1 och 2 då det är lättast att koppla in krokodilklämmor där.

För att vara säker på att inte två (eller fler) LED lyser samtidigt måste vi se till att de som inte ska lysa är släckta i de olika villkoren. Det blir därför extra många block.



Vi skickar digitalt till LED vilket innebär På (1) och Av (0). Som du kan se i programmeringen finns det ingen möjlighet att två LED kan lysa samtidigt. Undersök gärna i simulatoren för att se att det stämmer.

Koppling:

Precis som i lektion 2 måste vi återkoppla till micro:biten via en pin, GND. Vi ska göra så även nu men med skillnaden att alla tre LED ska tillbaka till samma GND-pin.

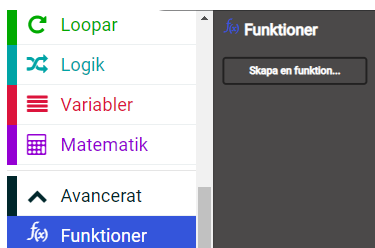


Lektion 5: Vandrande LEDs (disco).

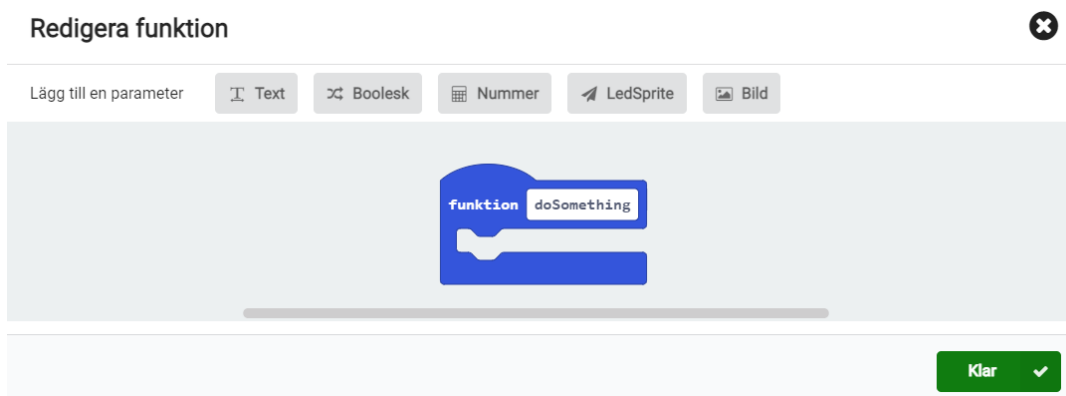
Uppdrag: Kan du göra en programmering med vandrande LEDs (discoblink)?

Om man ska upprepa en del av en programmering kan man använda loopar. Man kan också använda en funktion. En funktion kan ses som en annan sidoprogrammering som man enkelt kan lägga in i sin programmering med bara ett block.

Vi kommer att använda en helt ny blockrubrik, kallad "Funktioner".



Som ni kan se finns det inte så många block att välja på (faktiskt inga alls). Det beror på att vi först måste skapa vår funktion. Jag skapar en funktion som får heta "vandrande LEDs". Jag trycker då på "Skapa en funktion...":



Här kan man lägga till olika parametrar men vi ska bara göra en sak; där det står "doSomething" skriver vi "vandrande LEDs". Sen trycker vi på "Klar".



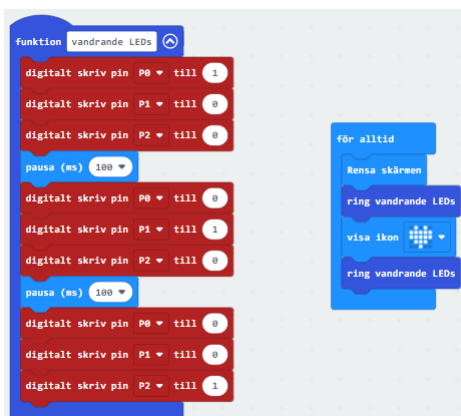
Efter vi skapat funktionen får vi block att jobba med. Först ett funktionsblock som läggs ut på programmeringsytan direkt och även två block under blockrubriken.

I funktionsblocket lägger vi de block som vi vill ska ingå i funktionen. Vi vill ju ha vandrande LEDs så vi kan använda de block vi hade i förra lektionen (enbart på- och av-blocken för våra LED). Vi lägger in de i funktionen.



För att få våra LED att vandra har jag lagt in två stycken vänta-block med 100ms (0,1s). Om man inte har dessa kommer funktionen köra igenom så snabbt att bara LED kopplad till pin 2 kommer lysa (sista blocket).

Nu när vi har vår funktion färdig kan vi använda oss av blocket "ring vandrande LEDs" så fort vi vill använda vår funktion i programmeringen. Vi kan också lägga till andra block. Ett exempel:



Huvudprogrammeringen ligger i "för alltid"-blocket och utan det händer ingenting. Funktionsblocket i sig är ingen "riktig" programmering som är körbar utan bara en funktion (definition) av vad som ska göras om man använder "ring vandrande LEDs"-blocket.

Det finns också ett ikonblock som visar ett hjärta och för att inte det ska visas hela tiden finns det även ett "rensa skärmen"-block som släcker skärmen (innan hjärtat kommer tillbaka igen).

Koppling:

Precis som i förra lektionen behöver vi koppla tillbaka alla korta ben (minus) till GND.

Testa att göra en egen funktion (eller två) och därefter en egen programmering.

MICRO:BIT

Utlåningslåda 1 - Grundläggande

Uppgifter

Vi har valt att dela upp innehållet i fem lektioner enligt följande:

1. Introduktion. Vad är en micro:bit? Hur programmerar man den?
2. Programmera figurer.
3. Skriv text och siffror.
4. Använda knappar.
5. Skakafunktion.

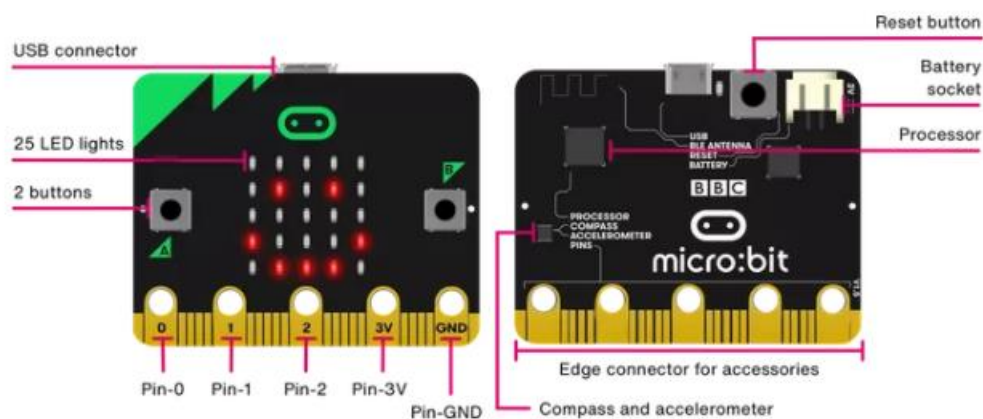
Nedan följer mer utförlig beskrivning av de olika lektionerna.

Lektion 1: Introduktion

Introduktionen finns i ett separat dokument. Där beskrivs hur man kopplar upp den mot en dator eller en iPad. Men vad är en micro:bit och hur fungerar den?

Micro:bit

En micro:bit är en typ av "enkorts dator". Det betyder att alla funktioner som finns i en dator finns, fast på ett kort.



En dator har följande som har sin motsvarighet på micro:biten enligt följande:

- Skärm. En dators skärm är till för att kunna visa information. Micro:biten har en skärm men den är nedbantad till en LED-variant med 25 st röda lysdioder.
- Tangentbord. På datorn använder vi tangentbordet för att mata in information. Micro:biten har inget "riktigt" tangentbord men man kan mata in information med hjälp av två knappar (tre kombinationer: A, B eller A+B). En tredje knapp är placerad på baksidan och har en enda funktion; den startar om micro:biten och den programmering som finns lagrad i minnet.
- Minne. En dator har minne i form av en hårddisk (för lagring) och ett arbetsminne (för det datorn arbetar med för tillfället). Micro:biten har båda minnena i ett och samma, ett s.k. flashminne. Begränsningen ligger i att det är ett väldigt litet minne och kan bara hålla en programmering i minnet åt gången. Däremot så sparas programmeringen även då micro:biten blir strömlös så den försvinner inte när du stänger av den (kopplar ur kabel till dator eller batteripack).
- Utöver de tre ovan har micro:biten något som kallas sensorer. Dessa är olika typer av komponenter som ger micro:biten kontinuerlig information. De som finns är:
 1. Ljussensor. Ger information om ljusstyrka. Ljussensorerna är placerade i LED-matrisen (vissa LED är dubbelriktade dvs. de kan både skicka ut ljus och ta emot ljus).
 2. Accelerometer. Ger information om hur micro:biten är vinklad. Håller du den i handen vet den därför om du vinklar den åt höger eller vänster (även mot och ifrån dig). Jämför med moderna smartphones som har inbyggd accelerometer för att känna av hur du vinklar din telefon.
 3. Temperatursensor. Ger information om temperaturen vid micro:biten.
 4. Kompass-sensor. Ger information om de magnetiska polerna precis som en vanlig kompass.

Dock kan vi inte se sensorernas information om vi inte i vår programmering uttryckligen talar om att vi vill se det.

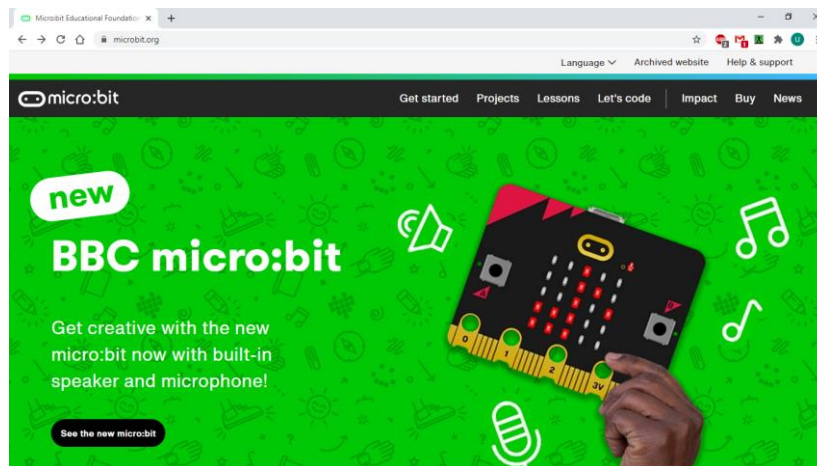
- Micro:biten har också en USB-anslutning för att koppla till dator via USB-kabel och en batterikontakt för att ansluta batteripack. Om man vill använda micro:biten med programmering utan att behöva ha den kopplad till en dator för att kunna få ström kan man använda batteripacket. Bredvid USB-kontakten på baksidan finns en liten gul LED som lyser när micro:biten får ström och blinkar när micro:biten tar emot information (när man laddar över programmeringar från dator till den).
- Om man vill använda en iPad/tablet (eller en iPhone) kopplar man upp micro:biten mot den utan kabel med hjälp av Bluetooth. Man kan även kommunicera mellan två (och fler) micro:bit med hjälp av radiovågor.

Allt det här finns alltså på det lilla kortet! Längst ner finns även ett kopparfält med siffror och hål. Dessa tar vi upp i en fortsättningskurs när vi kommer att koppla till andra saker till vår micro:bit.

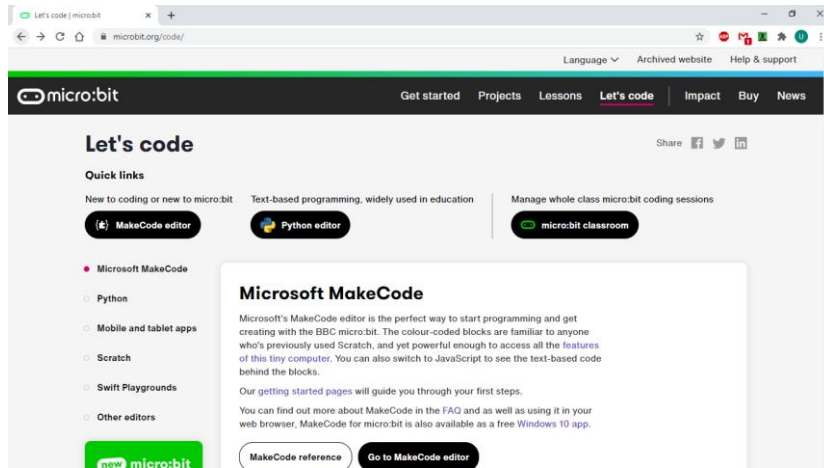
Hur programmerar man?

I instruktionen hur man gör för att koppla upp micro:biten fick du en liten inblick i vilket program som används vid programmering av micro:bit.

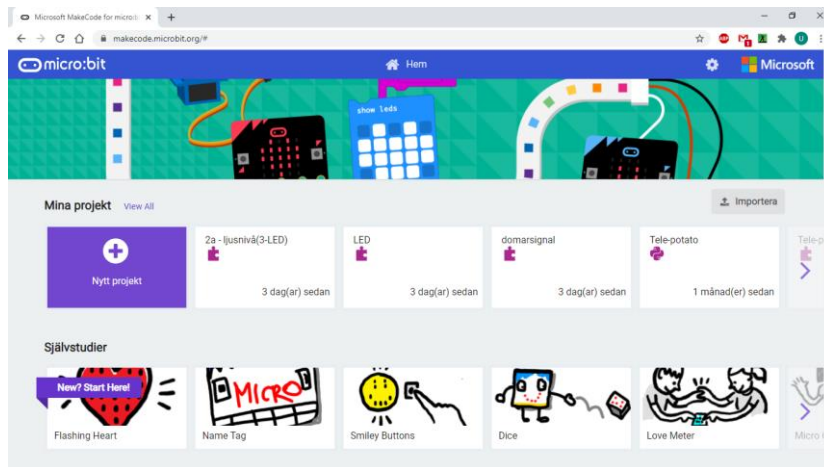
Gå in på <http://microbit.org/>



Koppla in din micro:bit med USB-kabeln och gå in på <http://microbit.org/code/> alternativt parkoppla din micro:bit med din iPad.

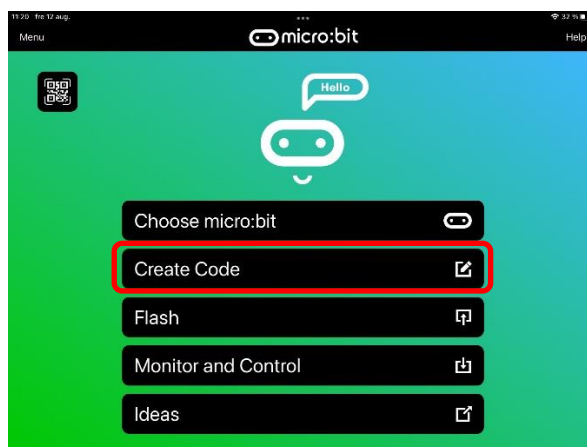


och under "New to coding or new to micro:bit", tryck på knappen märkt "MakeCode editor".

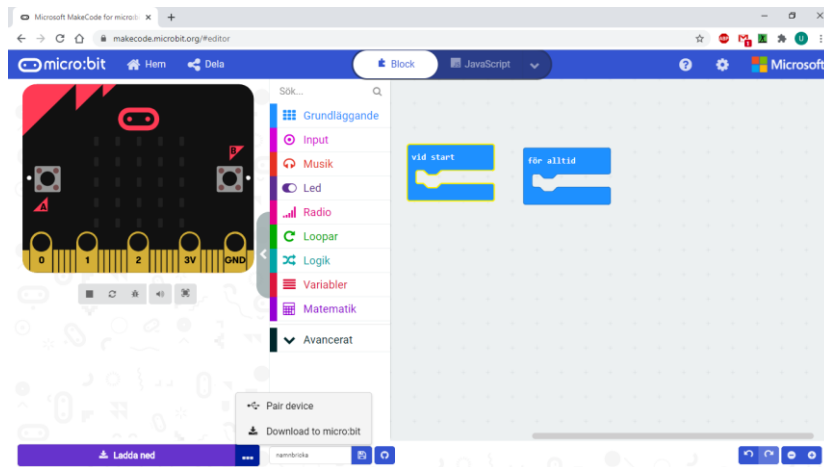


Det här är hemskrmen för programmering. Under "Mina projekt" sparas de programmeringar som du gjort tidigare. Är det tomt där beror det på att du inte gjort några programmeringar på den dator du sitter vid.

Använder du iPad så väljer du "Create Code" i appen.

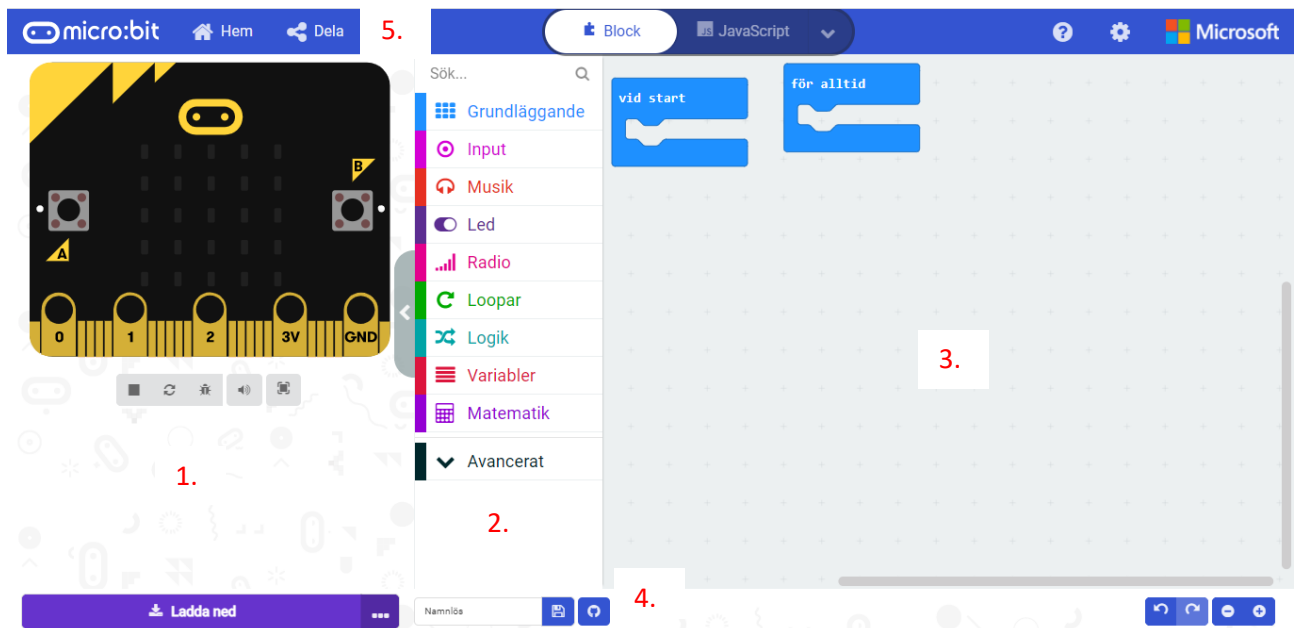


För att starta ett nytt projekt på datorn trycker du på "Nytt projekt".



Varje nytt projekt ser likadant ut. Du har två stycken block; "vid start" ("on start" på engelska) och "för alltid" ("forever" på engelska). Dessa två s.k. startblock kan vara bra när man startar sin programmering. Vill man inte ha dem kvar kan man radera dem genom att klicka på dem och dra dem till mitten av skärmen (där blockrubrikerna finns). Gör man det kommer det upp en stor papperskorg med röd bakgrund. Släpp blocket där så försvinner det.

Du behöver inte vara orolig att radera något. Alla block finns att hämta tillbaka under blockrubrikerna!



1. Simulator. Till vänster finns det en bild av en micro:bit. Den är till för att du ska kunna prova din programmering innan du skickar över den till micro:biten. Man kan säga att det är en virtuell micro:bit där knapparna fungerar som på den fysiska och skärmen visar information.
2. Blockrubriker. För att kunna programmera behöver vi olika block som gör olika saker. Dessa är organiserade under rubriker så att man lättare ska kunna hitta det man letar efter. De är även färgkodade för att det ska bli lättare att titta igenom sin programmering.

- Arbetsyta. Här lägger man de block som man vill ha med i sin programmering. Det är viktigt att man kopplar ihop blocken med varandra. Man kan enkelt se om ett block inte är ihopkopplat med ett annat block och med i programmeringen genom att det blocket blir gråmarkerat med små tunna gula streck.



- I nedre delen av skärmen finns knappar för att döpa sin programmering, spara, ladda ned till micro:bit men även ångra-funktioner och zoom (till höger).
- I övre delen av skärmen finns "Hem" som skickar dig tillbaka till hemskrmen (se tidigare bild). Längst till vänster står det "micro:bit". Trycker du där kommer du till microbit.org.

Lär känna micro:biten del 1:

[Uppvärmning](#)

Lektion 2: Programmera figurer

- Hjärta (på engelska)
<https://microbit.org/news/2020-04-24/microbit-at-home-heart/>

Uppdrag: Skapa en programmering som visar ett hjärta på micro:bitens skärm. Länken pekar på beskrivning hur man gör.

- Bultande hjärta (på engelska)
<https://microbit.org/projects/make-it-code-it/beating-heart/>

Uppdrag: Jobba vidare med föregående programmering och få hjärtat att bulta. För att få den effekten programmerar man in att skärmen växlar mellan ett litet och ett stort hjärta.

- Rörliga djur (på engelska)
<https://microbit.org/projects/make-it-code-it/animated-animals/>

Uppdrag: Skapa ett animerat djur på skärmen.

- Uppdrag: Designa ditt eget rörliga djur!

Lektion 3: Skriv text och siffror

- Namnbricka (på engelska)
<https://microbit.org/projects/make-it-code-it/name-badge/>

Uppdrag: Skapa en namnbricka som visar ditt namn. När micro:biten skriver text måste den rulla texten för att få med alla bokstäver. Det gör den automatiskt.

Lär känna micro:biten del 2 "Visa ditt namn" (på svenska):

[Visa Ditt Namn](#)

- Slå tärning (på engelska)
<https://microbit.org/projects/make-it-code-it/dice/>

Uppdrag: Skapa en tärning. En tärning fungerar ju som så att den slumpar ut vad den visar när vi slår den. Vi kan inte "slå en micro:bit" men vi kan göra en programmering som slumpar ut en siffra och på så vis skapa en tärning. Micro:biten har en inbyggd accelerometer vilket gör att du kan skaka den.

- Visa temperatur (på engelska)
<https://microbit.org/projects/make-it-code-it/thermometer/>

Uppdrag: Micro:biten har en inbyggd temperatursensor. Skapa en programmering som gör vår micro:bit till en termometer. På sidan finns även hur man kan göra en ljusmätare ("Sunlight sensor").

Lektion 4: Att använda knappar

- "Knappar med känslor" (på engelska)
<https://microbit.org/news/2020-04-26/microbit-at-home-emotion-badge/>

Uppdrag: Använd micro:biten så att den kan visa olika känslobilder utifrån vilken knapp du trycker på.

- "Rätt eller fel"

Uppdrag: Skapa en programmering så att du med hjälp av micro:biten kan visa "Rätt", t.ex figur :) eller √, eller "Fel", t.ex figur :(eller X med hjälp av A- och B-knappen. Då vi behöver kunna tömma skärmen (ta bort ikon/bilden) efter varje svar behöver vi inkludera det i programmeringen (förslagsvis genom att trycka på både A- och B-knappen samtidigt).

- Räknare (på engelska)
<https://microbit.org/projects/make-it-code-it/counter/>

Uppdrag: Skapa en räknare. I denna programmering ska du skapa en räknare som kan användas för att hålla koll på ett antal. Du kommer att få bli bekant med något som heter "Variabel". En variabel kan ses som en behållare där vi kan lagra något (ett minne). Det kan vara både text och siffror. Här lagrar vi en siffra och varje gång vi trycker på knappen så ökar vi variabelns värde med ett (+ 1).

Vill vi nollställa räkneverket (variabeln) så trycker vi A+B (dvs. båda knappar samtidigt).

- Uppdrag: Hitta på en egen programmering med egna figurer och byt vad knapparna styr.

Lektion 5: Skaka

- Sten, sax, påse (på engelska)
<https://microbit.org/projects/make-it-code-it/rock-paper-scissors/>

Lär känna micro:biten del 3 "Sten, sax, påse" (på svenska)
[Sten, sax, påse](#)

Uppdrag: Skapa spelet sten, sax, påse på micro:biten. Här används slumpantal för att kunna slumpa ut utfallet.

- Stegräknare (på engelska)
<https://microbit.org/projects/make-it-code-it/step-counter/>

Uppdrag: Gör en stegräknare av din micro:bit. Med hjälp av den inbyggda accelerometern kan man använda micro:biten som en stegräknare. Det enda som behövs är en programmering. Variabel behöver användas då antalet steg måste sparas allt eftersom.

- Uppdrag: Använd det du lärt dig och hitta på någon egen spännande programmering. Visa micro:biten med din programmering för en kompis. OBS! Visa inte din programmering. Låt kompiserna fundera ut hur din programmering ser ut.

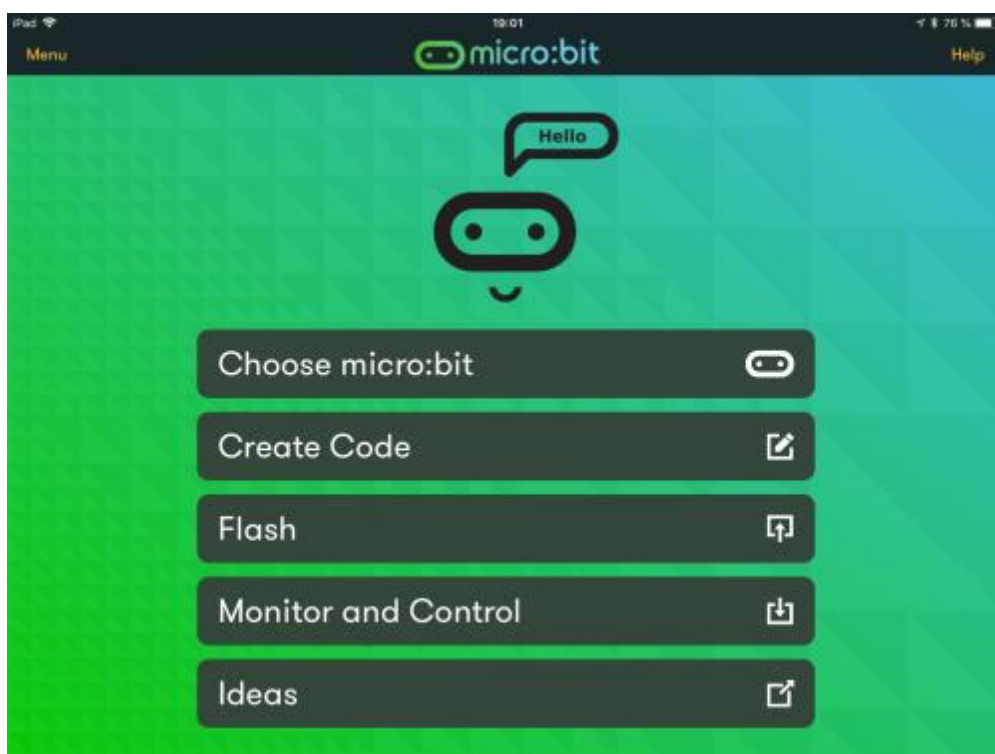
micro:bit – Ansluta micro:bit till iPad eller iPhone

Om du använder iPad eller iPhone finns det ett verktyg där du kan ansluta dem till en micro:bit. Anslutningen sker via bluetooth och låter dig programmera och läsa av värden från anslutningarna på din micro:bit. Appen ser ut så här i app store.



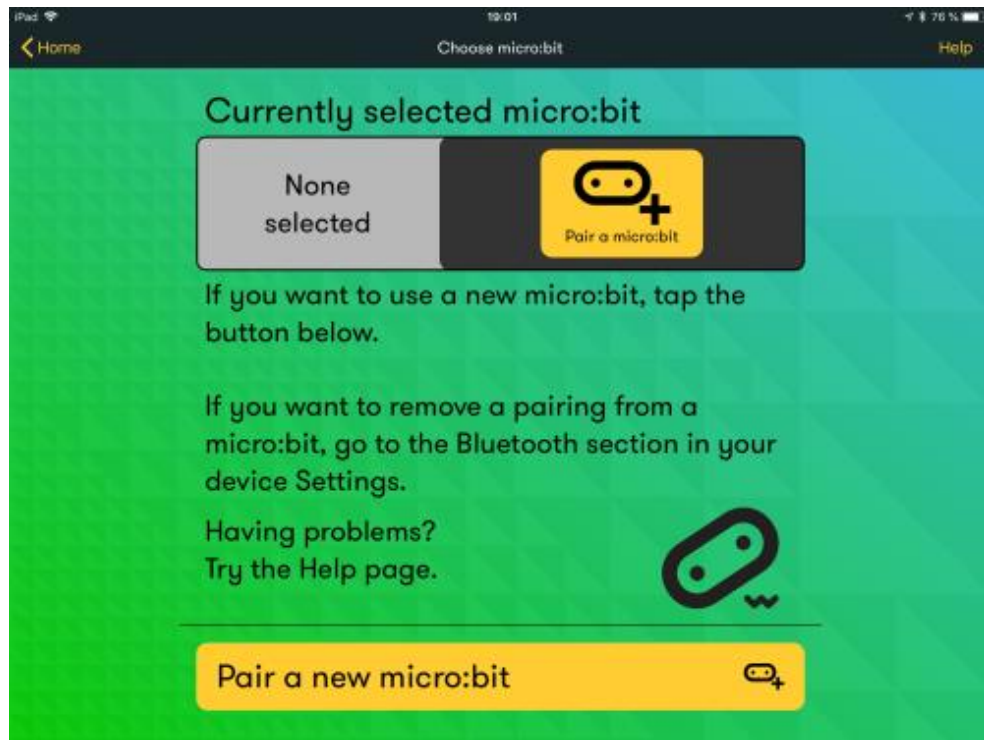
1.

Första gången du startar appen väljer du Choose micro:bit.



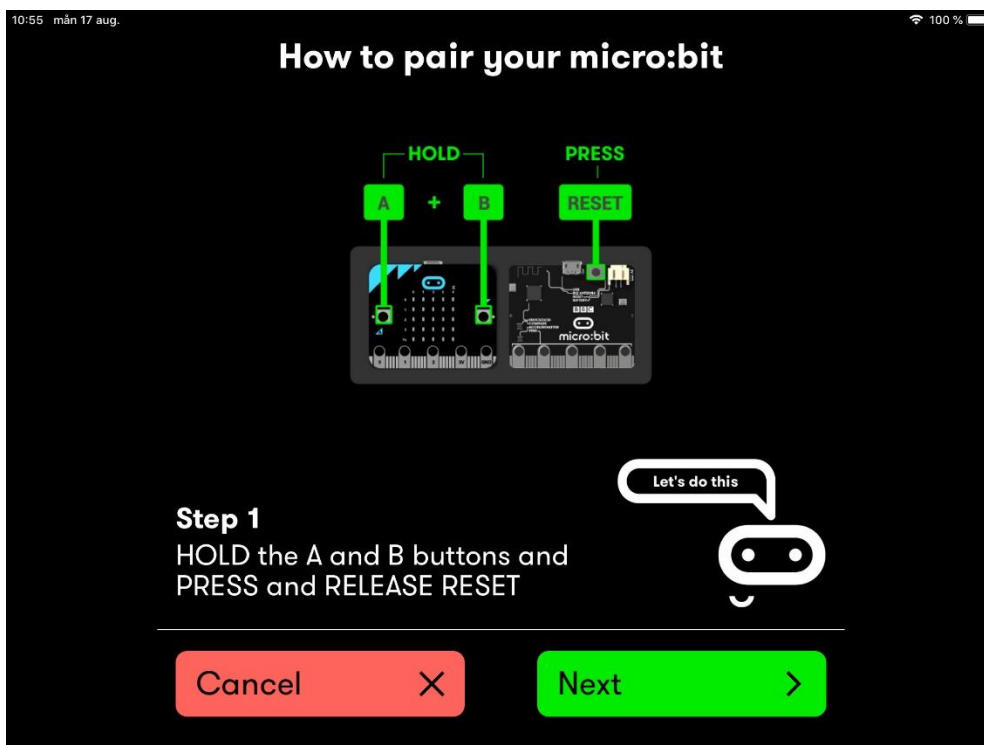
2.

På sidan som öppnas väljer du Pair a new micro:bit.



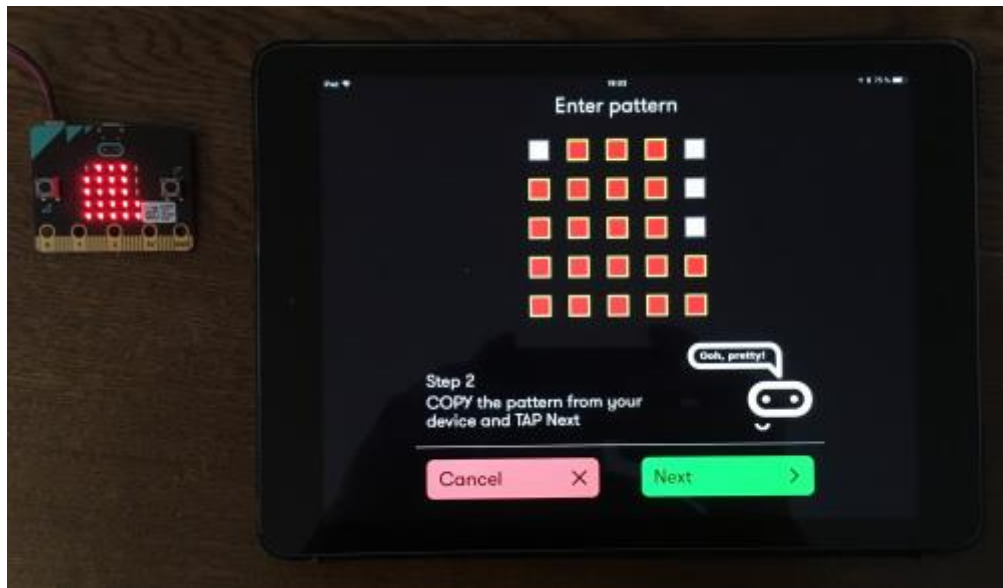
3.

Nu kommer ett steg som kräver lite fingerfärdighet. Håll nere knapparna A och B på din micro:bit. Medan du håller dem nere trycker du ner och släpper upp Reset på andra sidan av din micro:bit.



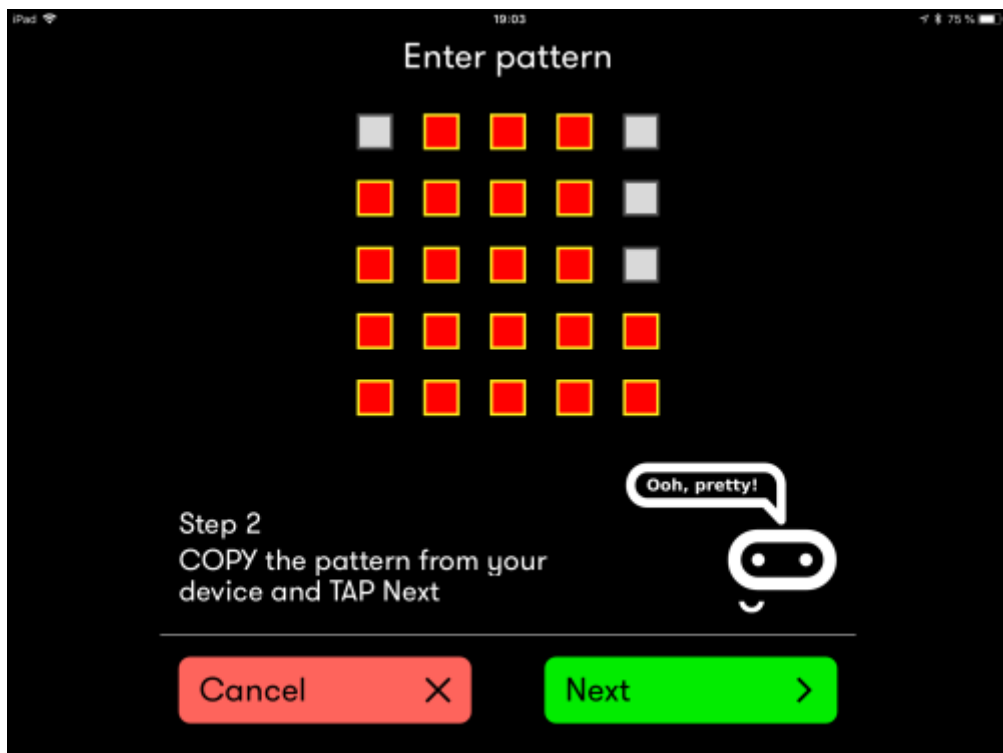
4.

Om du gjort rätt visar sig fem olika staplar på displayen mellan A och B-knapparna på din micro:bit.



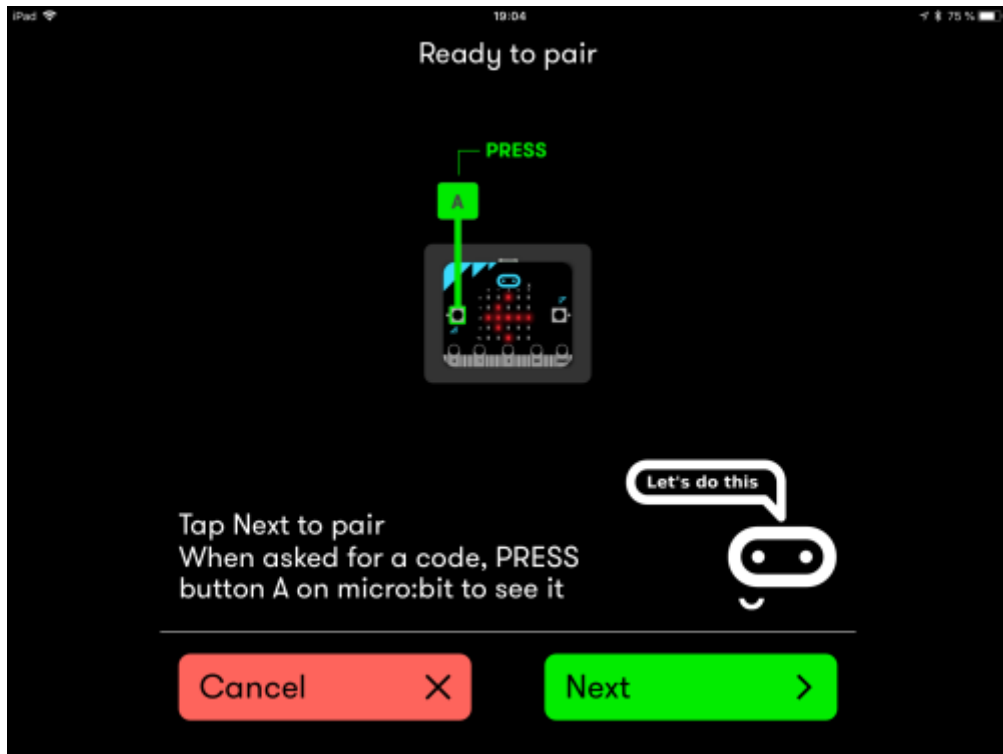
5.

Ställ in staplarna som visas i appen på samma sätt som de visas på din micro:bit och tryck på Next.



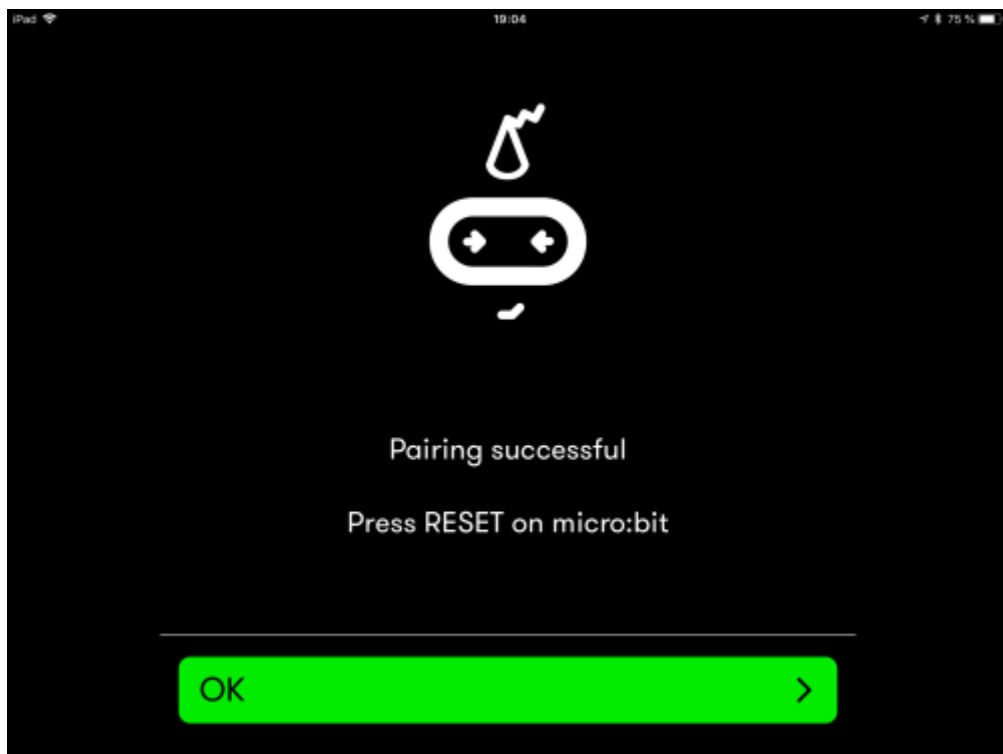
6.

På nästa sida står det att du ska trycka på A när appen frågar efter en kod. Det behöver du inte göra. Tryck bara på Next.



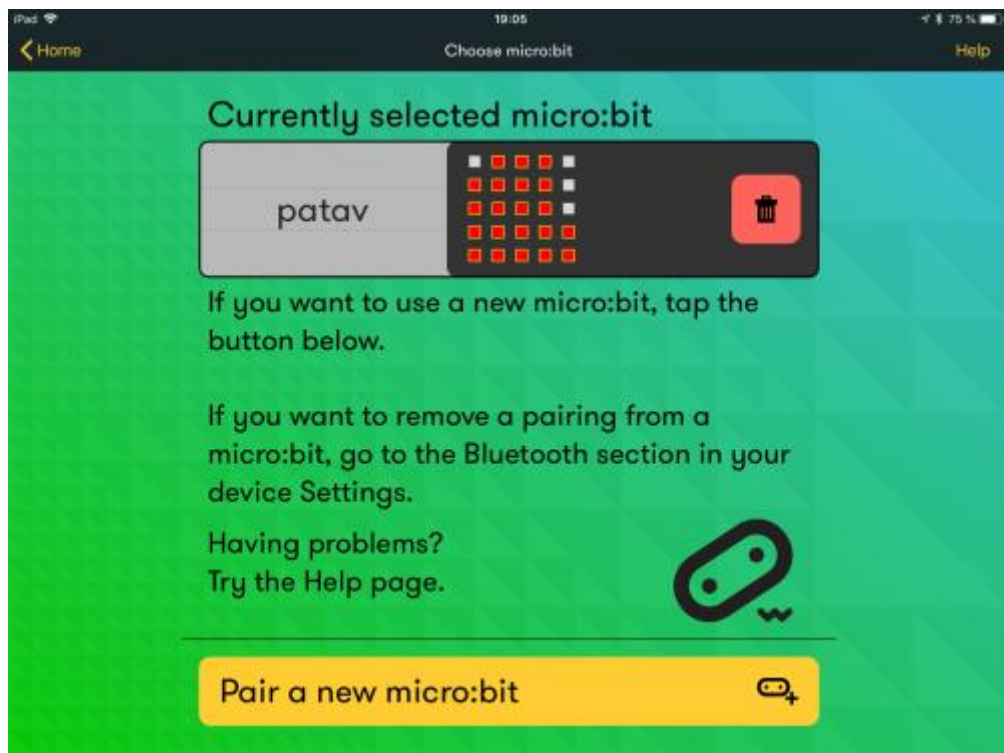
7.

Du kan få en fråga om Bluetooth Pairing Request. Välj i så fall Pair. Om allt gått bra får du meddelandet Pairing successful. Tryck på OK.



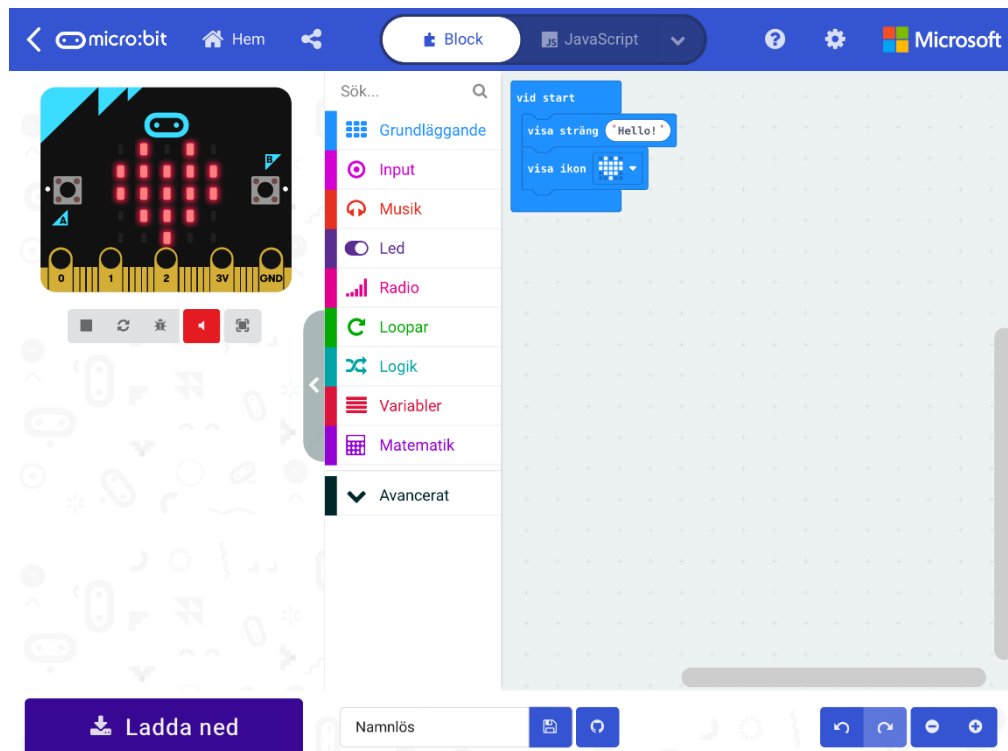
8.

Nu visas en sida där det står Currently selected micro:bit och under det visas mönstret som du tryckte in. Nu kan du klicka på Home högst uppe till vänster och fortsätta med programmering.



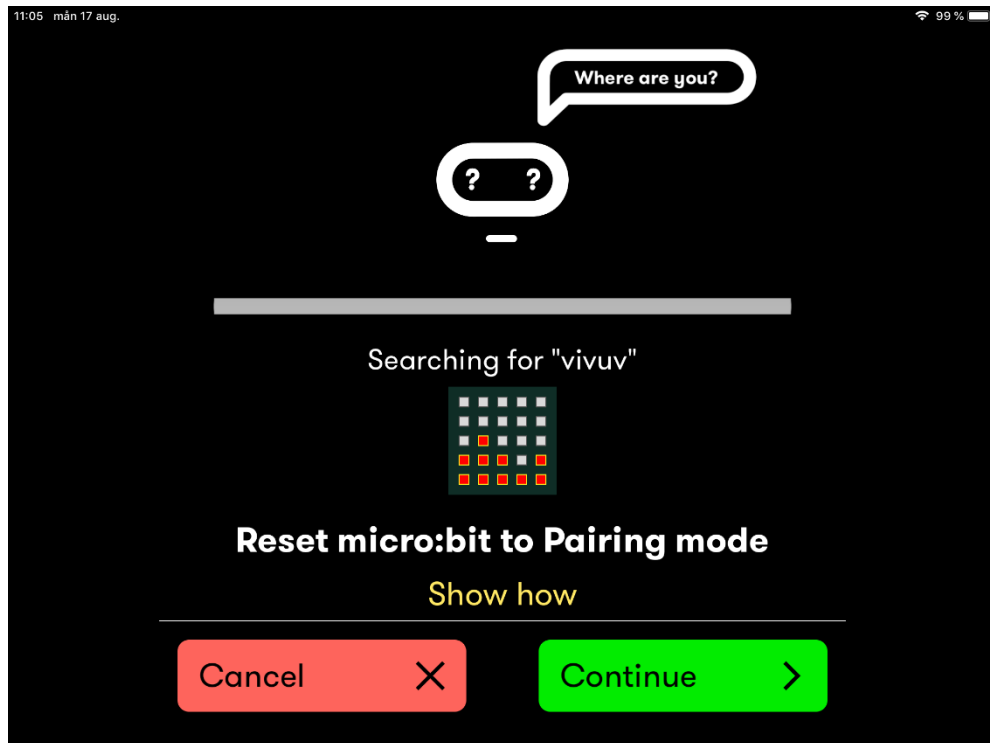
9.

Efter du gjort din programmering vill du överföra den till micro:biten.



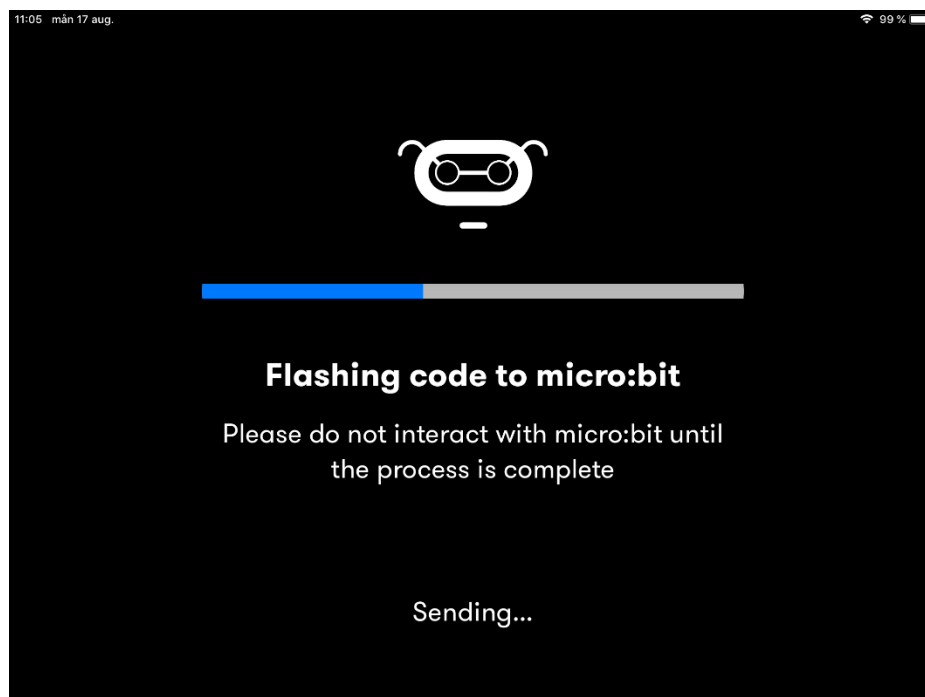
10.

Tryck "Ladda ned" och du kommer till nästa skärm (notera att det är en annan micro:bit nu, kallad "vivuv" isf. "patav").



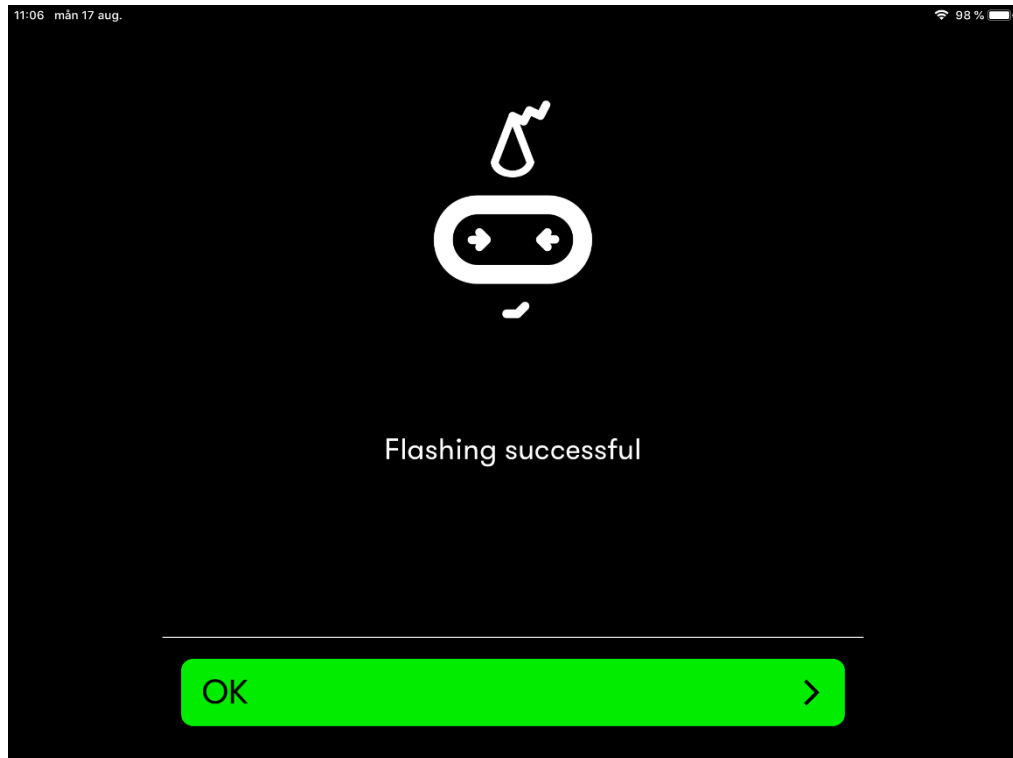
11.

Programmeringen kan inte överföras för att iPaden inte hittar micro:biten. För att lösa detta, tryck på A + B + Reset samtidigt på micro:biten. Släpp därefter Reset och håll kvar tills micro:biten visar bluetoothikonen på skärmen. Därefter tryck på "Continue" på iPaden. Programmeringen överförs.



12.

När överföringen är klar ser du denna skärm och programmeringen fungerar på din Pad.



13.

Man kan med fördel trycka A + B + Reset på micro:biten precis innan man trycker på "Ladda ned" på iPaden (bild 10). Då överförs programmeringen direkt.