För dig som vill undervisa i programmering på ett roligare sätt

Ordbok & instruktioner

micro:bit

Av grupp 9

Innehållsförteckning

Om micro:bit

Micro:bit är en utbildningssatsning från BBC i England, med målet att motivera och inspirera unga människor att utveckla ett teknik- samt programmeringsintresse (**http://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit**, The BBC micro:bit) (8 maj 2017).

Micro:bit är en liten programmerbar dator som kan hållas i handen. Den ges ut gratis till alla 11-åriga skolungdomar i Storbritannien. Den har utvecklats från BBC:s original micro:bit som kom ut under 80-talet och är 70 gånger mindre samt 18 gånger snabbare än dess föregångare. (**https://www.microbit.co.uk/about**, All the bits that make up your micro:bit) (8 maj 2017).

Micro:bit har två stycken knappar i höjd med varandra på framsidan. Knapp A som är vänsterknappen, och knapp B som är högerknappen. (**http://microbit-micropython.readthedocs.io/en/latest/button.html**, BBC micro:bit micropython) (8 maj 2017). De båda knapparna kan, när de trycks in var för sig eller samtidigt, aktivera vald funktion. Framsidan har också 25 stycken röda led-lampor som kan visa meddelanden som användaren programmerar in.
(**https://microbit.org/hardware**, Hardware: all the pieces that make up your micro:bit) (9 maj 2017).

Kort om programmering

Programmering är ett sätt att tala om för en dator i ett program vad den ska göra. Man programmerar genom språk som datorn kan översätta till maskinkod, programmering är som ett sätt att prata med datorer. Det finns flera sätt att programmera på och det finns flera olika språk man kan programmera på; C-språket av Microsoft, Python och Java av Oracle är bara några exempel. Men vad som är gemensamt för nästan alla programmeringsspråk är att de är skrivna på engelska. Alla begrepp ni kommer använda er av kommer att vara på engelska så det är viktigt att ni inte översätter dem direkt till svenska utan har kvar det engelska namnet men förklarar på svenska. I detta dokument finns förklaringar och definitioner för de grundläggande begreppen som behövs för en nybörjare att börja programmera, i detta fall anpassat för en ***micro:bit***.

Det finns flera liknande sätt att lära sig programmering som ***micro:bit***. Ett sätt är genom att koda med block. Istället för att se hela koden så ser man kodblock i olika färger och former som visar hur man, likt ett pussel, kan sätta ihop bitar av kod och skapa enkla program. Vad ***micro:bit*** gör är att den lägger till ett fysiskt element till programmeringen som annars bara sker digitalt. Eleverna programmerar ***micro:biten*** och själva chippet fungerar som en minidator, micro:biten utför funktionerna som kodblocken anger. Att programmera med block är lättare än att programmera direkt med kod, och **micro:bit** har tillräckligt med kodblock upp till en avancerad nivå. Om de rätta kodblocken saknas kan programmen annars bli begränsade.

Denna instruktion baserar sig på programmering med hjälp av JavaScript som trots namnet **inte är samma sak** som **Java-**pro-

grammering. **JavaScript** är ett webbaserat språk som är gjort för att få webbplatser mer interaktiva med script som aktiveras av användarens interaktion. Men det är fortfarande ett programmeringsspråk. ***Varför just JavaScript?*** JavaScript är ett simpelt språk som är enkelt att lära sig, och är därmed perfekt för nybörjare. JavaScript har man även användning för vid anpassning av en webbplats.

Länkar

Här är micro:bits hemsida: **http://microbit.org/about/**
Där kan du även hitta färdiga lektionsplaneringar som du kan använda till era lektioner: **http://microbit.org/teach/**
Här kodar du: **http://microbit.org/code/**

Här är vår hemsida som underlättar för dig som lärare:
http://student.mtstudent.se/~sh16hp2560/microbit/

Generellt och syntax

Micro:bits kodblock och skriven kod läses uppifrån och ner från höger till vänster. Rad för rad. Detta gäller majoriteten av andra programmeringsspråk också. Så fundera först på vad den första funktionen bör vara för att nästa steg i programmet ska ske. Logik och matematik är verktyg som är bra att ha till hands när man programmerar. ***Vad behövs***?
Skapa ett flödesschema där du med hjälp av block i olika former visar och kort beskriver vad koden ska göra samt i vilken ordning den ska köras. Detta underlättar och är bra att ha om man skulle fastna i planeringen. Det finns mycket resurser på Internet om flödesscheman samt verktyg för att skapa dem, till exempel programmet Draw.io.

Om du till exempel har ett program som ska slumpa fram ett tal från en lista behöver du först skapa tal, sedan skapa en lista och sedan lägga in talen i listan.
Skapa sedan en slumpgenerator och konfigurera den sedan så att den slumpar fram ett tal ur listan.

Begrepp att komma ihåg:

variabler
Variabler är behållare som du kan skapa och ge egenskaper samt fylla med innehåll.
I **Micro:bit** kan man skapa variabler i form av kodblock och i en egenskriven kod. Variabler är bra att ha när man vill spara information på ett visst sätt som man sedan kan använda i senare funktioner och uträkningar. Namnge variablerna så att man lätt kan skilja på dem och veta vad de innehåller bara på namnet.

funktioner

Inom programmering är en funktion, även subrutin, procedur, metod, underprogram eller subprogram, en del av ett datorprogram som kan anropas för att utföra en viss uppgift oberoende av resten av koden.

Kom igång

Du börjar med att öppna micro:bits hemsida för att koda och där skapar du dina program. För att föra över dina program till din ***micro:bit*** kan du koppla på två sätt: via sladd eller Bluetooth. Sladd är lättare och kan därför rekommenderas.

Via sladd: För att föra över filen du har gjort kod ifrån ***micro:bits*** webbplats till micro:biten behöver man först filen, den gör man på datorn exempelvis genom ***micro:bits*** egna editor. Man sparar ner filen till en valfri plats på datorn, vi rekommenderar att man gör en mapp för ***micro:bit.*** Sedan kopplar man in ***micro:biten*** genom dess usb-portal vid övre kanten till datorns usb-port och drar över den tidigare nedsparade filen. Viktigt att komma ihåg är att nästa gång du kopplar in ***micro:biten***, raderas allt innehåll på den.

Koppla till bluetooth:

1. Sätt på bluetooth på mobilen.
2. Öppna upp micro:bit-applikationen och tryck på Connection i menyn.
3. Tryck på pair with a new micro:bit.
4. Håll in knapp A och B och tryck sedan på reset-knappen medan A och B fortfarande är intryckta. Håll knapp A och B intryckta tills PAIRING MODE! visas på skärmen. Tryck på next.
5. Tryck in mönstret i applikationen för att efterlikna det som visas på micro:biten. Tryck på next i applikationen.
6. När du blir tillfrågad efter en kod tryck på knapp A på micro:biten för att se den.
7. Skriv in koden du ser på micro:biten vid förfrågan. Om du har lyckats koppla mobilen så visas en bock på micro:biten.
8. Tryck på reset-knappen för att börja använda micro:biten.

Micro:bit kodblock och begrepp
BASIC

show number

Visar numret inskrivet i pusselbiten på LED-skärmen.
Kod: basic.showNumber()

show LEDs

Lyser upp lamporna där de är röda (tryck på dem med musen). Punkterna agera lamporna i dess position.
Kod: basic.showLeds(`
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
`)
Ändra . till # för tända en LED-lampa

show icon

Lyser upp lamporna enligt förvalt mönster i högra rutan på kodblocket.
Exempelkod: basic.showIcon(IconNames.Heart)

show string

Visar text inskriven i rutan i pusselbiten på LED-skärmen ett tecken i taget. (fungerar bara med engelskt text, inget å,ä,ö)
Kod: basic.showString(“text”)

forever

Håller kodblock aktivt så länge det sitter ihop med forever
Kod: basic.forever(() => {
kodblock

})

pause

Pausar nuvarande händelse i kontakt (ihoppusslat till) i antal millisekunder (ms)
Kod: basic.pause(100)

on start

Aktiveras först och när micro:biten startas
Kod: vad som skrivs först uppifrån och ner

clear screen

Släcker LED-lamporna
Kod: basic.clearScreen()

show arrow

Visar en pil uppbyggd av LED-lamporna tända i olika formationer. Namgivna efter väderstreck för att underlätta riktning av pilen. T.ex. North pekar pilen uppåt, SouthWest pekar pilen nedåt till vänster etc.
Kod: basic.showArrow(ArrowNames.North)

INPUT

on button [...] pressed

Kodblock som aktiverar kod när vald knapp är nedtryckt.
Knapp A-Kod: input.onButtonPressed(Button.A, () => {

})

on [...]

Kodblock som aktiveras när den som håller micro:biten rör den på ett speciellt sätt. t.ex skakar den
Kod: input.onGesture(Gesture.Shake, () => {

})

on pin [...] pressed

Kodblock som aktiveras när användaren trycker på pin P0 (0), P1 (1) eller P2 (2) (de runda på nedre delen av micro:biten).
Kod: input.onPinPressed(TouchPin.P0, () => {

})

button [...] is pressed

Kodbit för att kontrollera om A,B eller båda knapparna samtidigt har tryckts ned.
Kod: input.buttonIsPressed(Button.A)

pin [...] is pressed

Kodbit för att kontrollera om P0,P1 eller P2 har tryckts ned.
Kod: input.pinIsPressed(TouchPin.P0)

acceleration (mg)

Mäter accelerationen micro:biten utsätts för i verkligheten i X-led, Y-led, Z-led eller gentemot gravitiationen i tusendelar (mg)
Kod: input.acceleration(Dimension.X))

light level

Mäter ljusnivån runt micro:biten (alltså i rummet)
Kod: input.lightLevel()

compass heading (°)

Räknar ut vilket håll (i grader utifrån var magetometern pekar) micro:biten pekar.
Kod: input.compassHeading()

temperature (°C)

Mäter temperaturen kring micro:biten.
Kod: input.temperature()

rotation (°)

Mäter rotationen uppochnedåt (Pitch) eller åt sidorna (Roll) i grader.
Kod: input.rotation(Rotation.Pitch)

magnetic force (μT)

Mäter hur mycket magnetisk kraft micro:biten i vald riktning X-led (höger-vänster), Y-led (bakåt och frammåt) och Z-led (upp-ned).
Kod: input.magneticForce(Dimension.X)

running time (ms)

Mäter hur långt det har gått ett program har körts i millisekunder (ms)
Kod: input.runningTime()

on pin [...] is released

Kontrollerar om bestämd pin, P0, P1 eller P2, är släppt från att ha varit nedtryckt. Denna bit fungerar alltså som ett villkor.
Kod:
input.onPinReleased(TouchPin.P0, () => {

})

set accelerometer range

Ställer in delen av mikro: biten som mäter accelerationen (hur mycket mikrobitet går fort eller saktar ner), om du behöver mäta hög eller låg acceleration. 1g, 2g, 4g eller i 8g.
Kod: input.setAccelerometerRange(AcceleratorRange.OneG)

MUSIC

play tone [...] for [...] beat(s)

Spelar upp en vald ton (från lågt C to Högt B, likt ett piano) i antal beats/sekund.
Kod: music.playTone(175, music.beat(BeatFraction.Eighth))

ring tone (Hz)

Ringer en ton så länge som programmet är igång. Välj på samma sätt som föregående tonväljare.
Kod: music.ringTone(262)

rest (ms)

Skapar ett uppehåll i musiken för vald tid i slag (beats).
Kod: music.rest(music.beat(BeatFraction.Breve))

start melody [...] repeating [...]

Spelar en vald standard melodi i vald tid (en gång, för evigt, en gång i bakgrunden eller för evigt i bakgrunden)
Kod: music.beginMelody(music.builtInMelody(Melodies.Funk), MelodyOptions.Forever)

tempo

Mäter tempot av det pågående "musiken" som spelar i bpm (beats per minute).
Kod: music.tempo()

change tempo by (bpm)

Ändrar tempo på musik till valt antal bpm.
Kod: music.changeTempoBy()

set tempo by (bpm)

Bestämmer tempot till valt antal bpm.
Kod: music.SetTempo()

LED

plot och unplot

Lyser upp en av LED-lamporna enligt koordinaterna inskrivna i x och y-led. 0,0 är lampan längst upp till vänster.
Kod: led.plot(0, 0)

toggle

"Togglar" en LED-lampa, sätter på den om den är avstängd eller stänger av den om den är på, också enligt samma koordinatsystem (x och y-led).
Kod: led.toggle(0, 0)

point

Kontrollerar ifall en LED-lampa är på eller inte på en koordinat, x och y-led.
Kod: led.point(0, 0)

plotbargraph -of -up to

Visar en bargraf av numrena du bestämmer. Visar numret som linjer med olika längder.
Kod: led.plotBarGraph(
    10,
    14
)

set brightness

Bestämmer ljusstyrkan på LED-lamporna enligt din specifikation. 0-255 där 0 är släckt och 255 är som ljusast.
Kod: led.setBrightness(255)

led enable

Bestämmer om LED-lamporna får lysa eller inte. Sant (true) eller false (falskt).
Kod: led.enable()

stop animation

Stoppar alla animationer och de som väntar på att spelas upp.
Kod: led.stopAnimation()

RADIO

radio send number

Skickar ett nummer till andra micro:bits kopplade via radio.
Kod: radio.sendNumber(5)

radio send value [...] = [...]

Skickar en string och ett nummer till andra micro:bits kopplade via radio. Maxlängden för stringen är 12 tecken.
Kod: radio.sendValue(“name”, 0)

radio send string

Skickar en string till andra micro:bits kopplade via radio.
Kod: radio.sendString(“”)

on radio received

Kör ett program när micro:biten får in ett nummer eller en siffra över via radio. Fungerar som ett villkor. Kan ta emot;
Kod: radio.onDataPacketReceived(({receivedNumber: name}) => {
})

radio set group

Gör ett program med det grupp-ID du berättar för att skicka och ta emot via radio. En grupp är som en kabelkanal (en micro:bit kan bara skicka eller ta emot i en grupp i taget). Ett grupp-ID är som kabelkanal-numret.

Om du inte berättar för ditt program vilket grupp-ID som ska användas med den här funktionen, kommer det att räkna ut sitt eget grupp-ID. Om du laddar samma program på två olika micro: bitar, kommer de att kunna prata med varandra eftersom de kommer att ha samma grupp-ID.

Kod: radio.setGroup(1)

radio set transmit power

Bestämmer sändningsstyrkan på radion
Kod: radio.setTransmitPower(7)

radio set transmit serial number

Gör radiopaketet inbäddat i serienumret med varje paket med data.
Kod: radio.setTransmitSerialNumber(true)

radio write received packet to serial number

Skriver det sista paketet som mottagits av radion till seriell i JSON-format. Ska kallas inom en återuppringning till det mottagna datapaketet.
Kod: radio.writeReceivedPacketToSerial()

1 Ett objekt med oordnad mängd data med nycklar **http://www.json.org/json-sv.html**

LOOPS

repeat [...] times do

Kör en del av ett program enligt dina antal gånger.
Kod: for (let i = 0; i < 4; i++) {

}

while [...] do

Kör ett program så länge villkoret att det körs, som användaren bestämmer, är true.
Kod: while (false) {

}

for [...] index from 0 to [...] do

Kör en del av ett program enligt dina antal gånger. Du kan dock sätta in en egen variabel istället för “index”.
Kod: for (let index = 0; index <= 4; index++) {

}

LOGIC

if [...] then

Kör ett program eller inte baserat på boolean; true eller false.
Kod: if (true) {

}

if [...] then else

Kör ett program eller inte baserat på boolean; true eller false. Men detta kodblock har ett else-element som körs om föregående villkor inte uppfylld (är false).
Kod: if (true) {

}
else{

}

Sedan finns det flera villkors pusselbitar som man kan använda när man vill jämföra tal, räkna med tal, lägga till “and” och “or” som ett villkor till t.ex. en if-sats

VARIABLES

create variables

Du skapar en variabel, med eget namn, i ett kodblock som du kan använda i villkor i t.ex. if-satser.
Kod: let item = 0

set [...] to

Bestäm ett värde på vald variabel du har skapat. Sätt dit en pusselbit som passar t.ex. ett tal eller en bokstav. Till och med en uträkning.
Exempelkod ger värdet 0: item = 0

change [...] to

Ändrar värdet på variabeln.
Exempelkod ändra värdet till 4: item += 4

Ordbok & instruktioner

micro:bit

OM

En bok för dig som vill undervisa i programmering på ett roligare sätt ….

Av grupp 9