

Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

Material

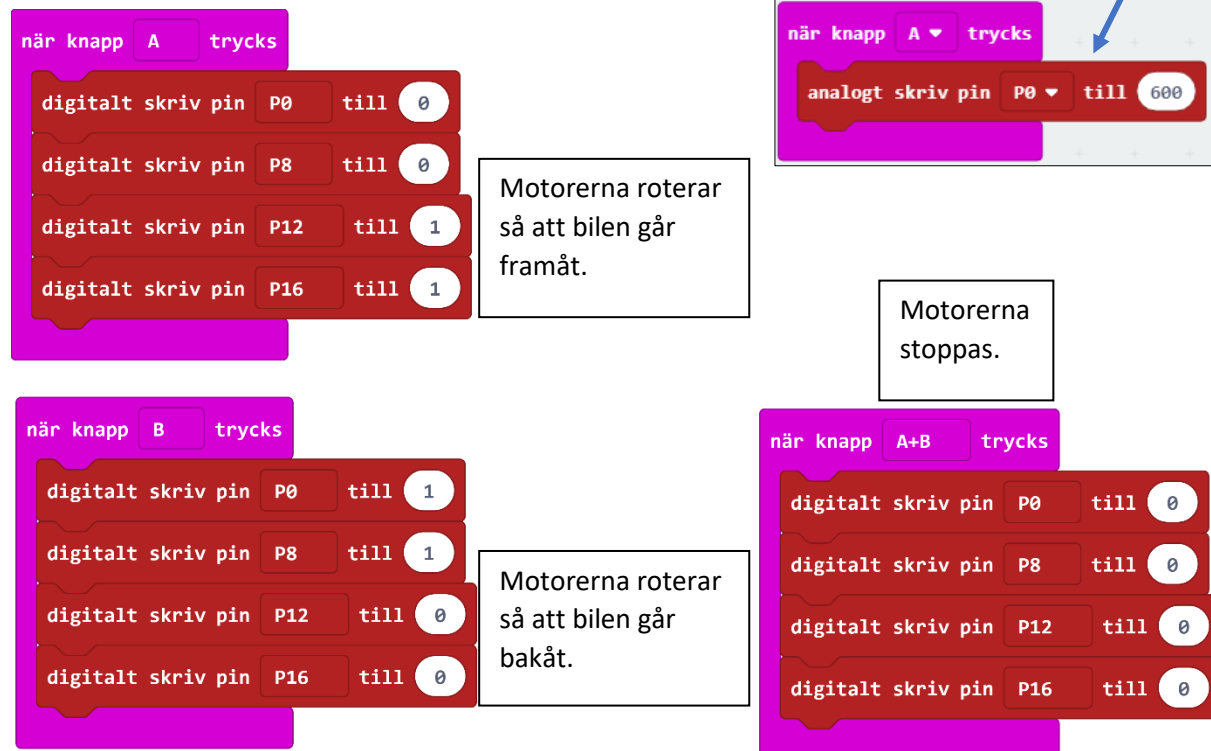
1. Micro:bit (2 st för radiostyrning), USB-kabel
2. Motorkort (Kitronik) + batterihållare (4 st 1.5 V AA)
3. 2 st. enklare DC-motorer, med kablar fastlödade
4. Kartong, plastkorkar, sugrör, grillpinnar, gummiband, remskivor
5. Limpistoler

Koppla motorerna till Microbit och Motorkortet. Batteripaketet kopplas till ingången på Motorkortet, var noga med svart till minus och röd till plus.

Testkod (Kolla så att motorerna går åt rätt håll, om inte byt polaritet på en av motorerna så att det stämmer med schemat). Du hittar knappblocken under **Input** och Pinblocken under **Avancerat > Pins**

PIN 0	PIN 8	PIN 12	PIN 16	EFFECT
OFF	OFF	OFF	OFF	The buggy is stationary
OFF	OFF	ON	ON	Buggy drives forwards in a straight line
ON	ON	OFF	OFF	Buggy drives backwards in a straight line
OFF	OFF	ON	OFF	Buggy turns left in the forwards direction
OFF	OFF	OFF	ON	Buggy turns right in the forwards direction
OFF	ON	OFF	OFF	Buggy turns left in the backwards direction
ON	OFF	OFF	OFF	Buggy turns right in the backwards direction

Om motorerna och bilen går för fort kan man reducera spänningen till motorerna med blocket [analogt skriv pin...] och testa med något värde mellan 0 – 1023.



Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

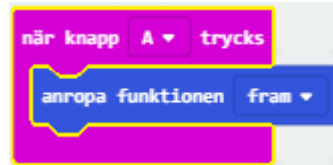
Underlätta livet med funktioner

Man kan skapa egna funktioner för att underlätta programmeringen av motorkortet.

Klicka på **Avancerat > Funktioner** och skapa en funktion som du kan döpa till **fram** (man kan använda åäö, men det är bra om vi undviker det. Engelska går utmärkt). Stoppa in Pinblocken och ställ in så att motorerna kör framåt. Gör fler funktioner med bakåt, höger – vänster och stopp.



Så här anropar du funktionen med ett tryck på knapp A:



Det betyder att vi inte behöver pussla en massa block varje gång vi ska ändra riktning, det räcker att göra en funktion för varje riktning.

Prova alla dina nya funktioner i en loop. Kom ihåg att ha en paus så länge som funktionen ska utföras.



Fungerar det som du tänk dig?

Svänger bilen för mycket? Du kan alltid ändra på tiden för att minska eller öka svängen.

Var det svårt att få tag på bilen? Den kommer att köra tills batterierna tar slut. På nästa sida löser vi så att vi kan programmera knapp A för att starta programmet och knapp B för att stoppa allt.

För att få stopp på motorerna i det här läget är det bra om du kommer åt batterierna, koppla loss ett så för du stopp. Eller dra ur micro:biten från motorkortet.

Om du jobbar på högstadiet och vill använda dig av textbaserad programmering så kan du växla till javascript eller använda micropython (codewith.mu eller python.microbit.org).

Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

En variabel hjälper oss att starta och stoppa bilen

Skapa en variabel och som i exemplet döp den till **start**. Ge variabeln värdet 0 och visa en ikon (krysset) som visar att vi stoppat allt när programmet börjar. Pussla ihop de olika kommandon till knapp A och knapp B. Det enda du behöver lägga i den koden är att sätta värdet på variabeln men bra att ha med olika ikoner också.

Ditt program för motorerna kommer ett nytt begrepp boolskt (boolean) villkor där **start = 1** men som också betyder True/Sant. Så om **start** är True/Sant (du har tryckt på knapp A) körs den koden hela tiden och om det är False/Falskt så kör vi den andra koden (dvs. stopp). En ganska enkel algoritm som löser vårt problem med start och stopp.

Här stoppar du in din kod för automatisk körning av bilen.

Du ser nu fördelen med funktionerna som du skapade tidigare.

Här stoppar du in din kod för att stoppa bilen.

Prova om du kan få till en kod som gör så att bilen lyckas flytta på eller vända alla toalettrullarna i utmaningen - bowling med bil.

Man kan givetvis använda sig av smarta lösningar innanför **[om <start> då]**. Kanske en loop som **upprepar ett antal gånger** för att sedan göra någonting annat i en annan loop. Du kan också använda loopen **[medan <sant> gör]** istället för **[om <start> då]**. Det finns obegränsade möjligheter att lösa problemet.

Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

Styra med radio (mer avancerat)

Du behöver två micro:bits (och ett batteri för den sändande micro:biten).

Programmera sändaren och mottagaren



OBS! Om det är flera grupper som använder radio samtidigt måste man separera dem i olika grupper.

Man kan använda gruppnummer upp till 255.

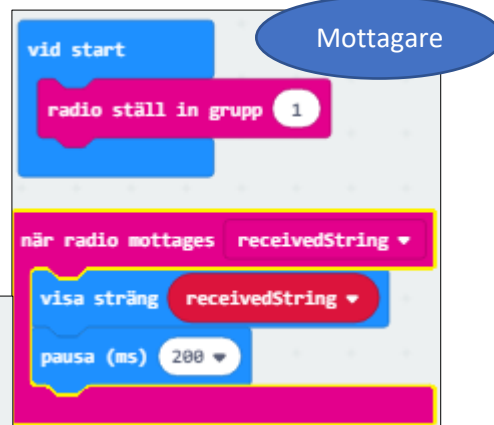
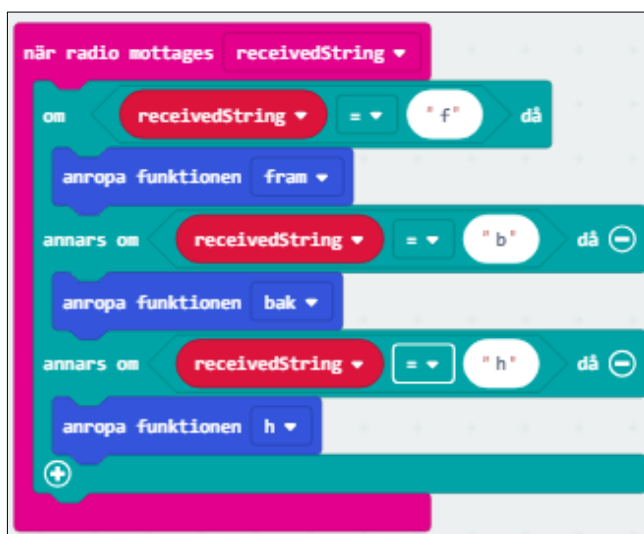
När man lutar den sändande micro:biten åt olika håll skickas en bokstav till mottagande micro:bit.

Mottagande micro:bit ska då förvandla koden så att bilen kör framåt, bakåt etc. enligt inkommande bokstav.

För att testa om kommunikationen fungerar gör du ett litet test för den mottagande micro:biten.

Fungerar det att ta emot bokstäverna?

Lycka till!



När det fungerar med överföringen fortsätter du med dina funktioner och lägger in dem i [när radio mottages...] flera if-satser (om – annars om – annars om...). Lägg till fler [annars om] med plustecknet.

Glöm inte att dina funktioner måste finnas innan du kan anropa dem.

Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

Samma kod men med micropython (textbaserat) istället

Du kan prova online-editorn python.microbit.org eller hämta MU-editorn på codewith.mu

1. Testa så att motorerna går åt rätt håll

```
from microbit import *
```

```
while True:
```

```
    if button_a.was_pressed():                # När knapp A trycks ned motorer framåt
        pin0.write_digital(0)
        pin8.write_digital(0)
        pin12.write_digital(1)
        pin16.write_digital(1)
    elif button_b.was_pressed():              # När knapp B trycks ned motorer bakåt
        pin0.write_digital(1)
        pin8.write_digital(1)
        pin12.write_digital(0)
        pin16.write_digital(0)
    elif button_a.is_pressed() and button_b.is_pressed(): #Knapp A + B stoppa motorerna
        pin0.write_digital(0)
        pin8.write_digital(0)
        pin12.write_digital(0)
        pin16.write_digital(0)
```

2. En funktion förenklar livet

```
from microbit import *                        # Samma som ovan fast med en funktion

def drive(pin):                              # En funktion som kan ha input (pin blir en lista)
    pin0.write_digital(pin[0])
    pin8.write_digital(pin[1])
    pin12.write_digital(pin[2])
    pin16.write_digital(pin[3])

while True:
    if button_a.was_pressed():
        drive((0, 0, 1, 1))                  # Anropa funktionen med 4 listvärden, obs. dubbla (())
    elif button_b.was_pressed():
        drive((1, 1, 0, 0))
    elif button_a.is_pressed() and button_b.is_pressed():
        drive((0, 0, 0, 0))
```

Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

3. Booleskt värde med variabel för start och stopp

```
from microbit import *
```

```
def drive(pin):                                # Funktionen drive(pin)
    pin0.write_digital(pin[0])
    pin8.write_digital(pin[1])
    pin12.write_digital(pin[2])
    pin16.write_digital(pin[3])

start = False                                  # Sätt ingångsvärde på variabeln start
display.show(Image.NO)                         # Visa ikonen (X)

while True:
    if button_a.was_pressed():                 # Om knapp A trycks så ge variabeln start värdet True
        start = True
        display.show(Image.YES)
    elif button_b.was_pressed():               # Om knapp B trycks så ge variabeln start värdet False
        start = False
        display.show(Image.NO)

    if start:                                  # Om start är True kör då sekvensen av kod
        drive((0, 0, 1, 1))
        sleep(2000)
        drive((0, 0, 1, 0))
        sleep(500)
        drive((0, 0, 0, 1))
        sleep(500)
        drive((1, 1, 0, 0))
        sleep(2000)
    else:                                       # Om start är False stoppa motorerna
        drive((0, 0, 0, 0))
```

Överkurs – om du hinner

Om du vill använda radiofunktionen och två microbit:s så kan du kolla koden på nästa sida. Två koder, en för sändande micro:bit och en för den mottagande micro:biten med motorkortet.

Principen är relativt enkel, där vi tar hand om lutningen av den sändande micro:biten för att skicka enkla textkommandon till mottagaren. Den inbyggda accelerometern kan ge värden i x, y och z lägen. Om vi lutar i x-led så får du ett positivt värde om micro:biten lutar åt höger (0 till 1023) om du lutar åt vänster får du ett negativt värde (0 till -1023). Samma värden kan du få i y och z-led.

Teknik – Styra egna konstruktioner – Micro:bit och Motorkort

1. Sändaren

```
from microbit import *  
import radio  
  
radio.on()  
radio.config(channel=1)  
  
while True:  
    if accelerometer.get_x() > 400:  
        display.show(Image.ARROW_E)  
        radio.send("right")  
        sleep(300)  
    elif accelerometer.get_x() < -400:  
        display.show(Image.ARROW_W)  
        radio.send("left")  
        sleep(300)  
    elif accelerometer.get_y() > 400:  
        display.show(Image.ARROW_N)  
        radio.send("forward")  
        sleep(300)  
    elif accelerometer.get_y() < -400:  
        display.show(Image.ARROW_S)  
        radio.send("backward")  
        sleep(300)  
    elif button_a.was_pressed():  
        display.show(Image.NO)  
        radio.send("stop")  
        sleep(300)
```

Importera biblioteket radio

Sätt på radiofunktionen
Konfiguera vilken kanal du vill köra på

Ett antal if-satser som tar hand om accelerometer
och om du lutar microbit:en skickar en textsträng

Om du trycker på knapp A skickas strängen **stop**

2. Mottagaren

```
from microbit import *  
import radio  
  
radio.on()  
radio.config(channel=1)  
  
def drive(pin):  
    pin0.write_digital(pin[0])  
    pin8.write_digital(pin[1])  
    pin12.write_digital(pin[2])  
    pin16.write_digital(pin[3])  
  
while True:  
    value = radio.receive()  
    if value == "forward":  
        drive((0, 0, 1, 1))  
    elif value == "backward":  
        drive((1, 1, 0, 0))  
    elif value == "stop":  
        drive((0, 0, 0, 0))
```

Importera biblioteket radio

Sätt på radiofunktionen
Konfiguera vilken kanal du vill köra på

Funktionen drive(pin)

Variabeln **value** får en textsträng som sändaren skickar.
I ett antal if-satser tar vi hand om värdet och startar
motorerna enligt de kommandon som sändaren skickar
Du kan lägga in för vänster - höger också med ytterligare
elif value == "right": och elif value == "left":