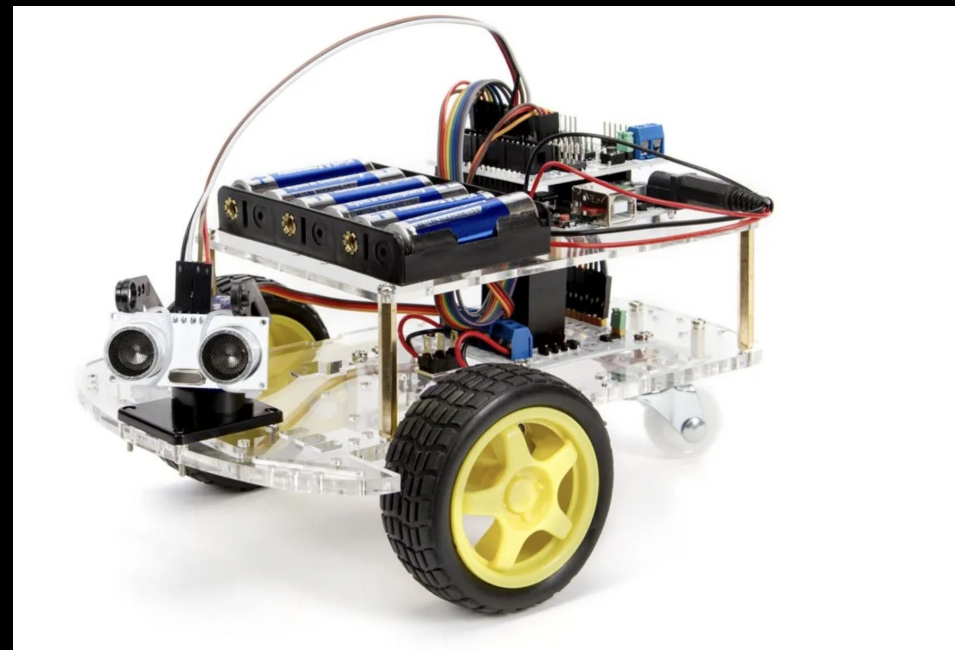


ARDUINO WORKSHOP

Selvkjørende robotbil

Agenda

- Kort om Arduino
 - Hva er Arduino?*
 - Hvorfor Arduino?*
 - Utbytte*
- Koding med C++
 - Bli kjent med kodespråket C++*
 - Arduino IDE*
- Dagens prosjekt
 - Fremgangsmåte*
 - Koden*



Arduino

- “Mikrokontroller” -> veldig liten datamaskin
- Leser innganger og gjør det om til utganger
- Enkelt og billig
- Gir et bredt spekter av muligheter

LED-cube

Lese av sensorer

Aktivere motorer

Selvkjørende roboter



Forstå brettet



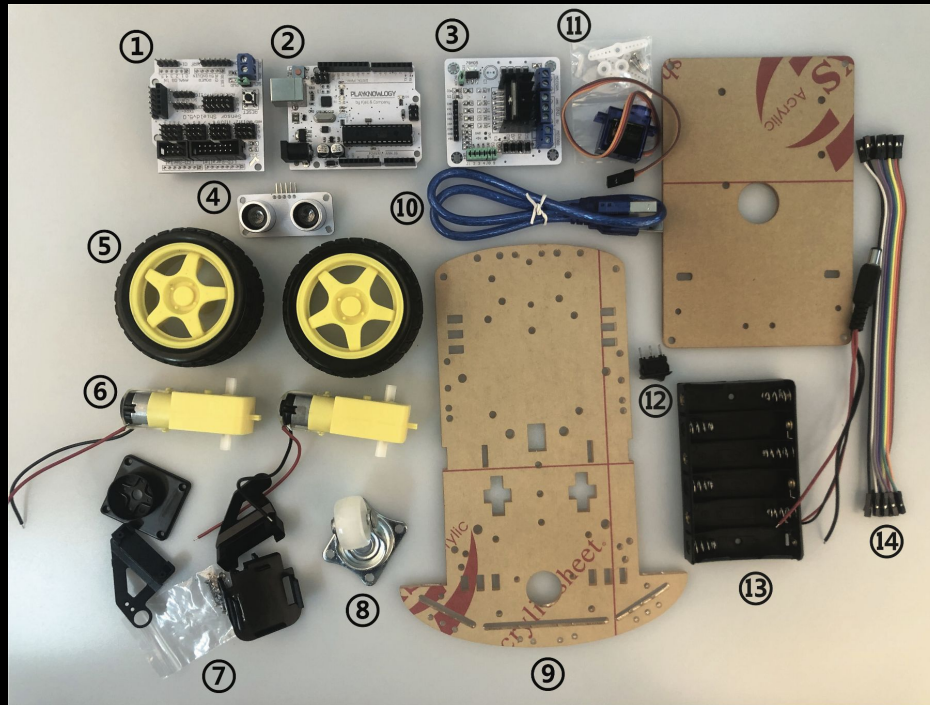
Arduino IDE

- Åpen kildekode
- Kode manipulerer arduinoen til å gjøre ønskede oppgaver
- Språket brettet forstår er C++



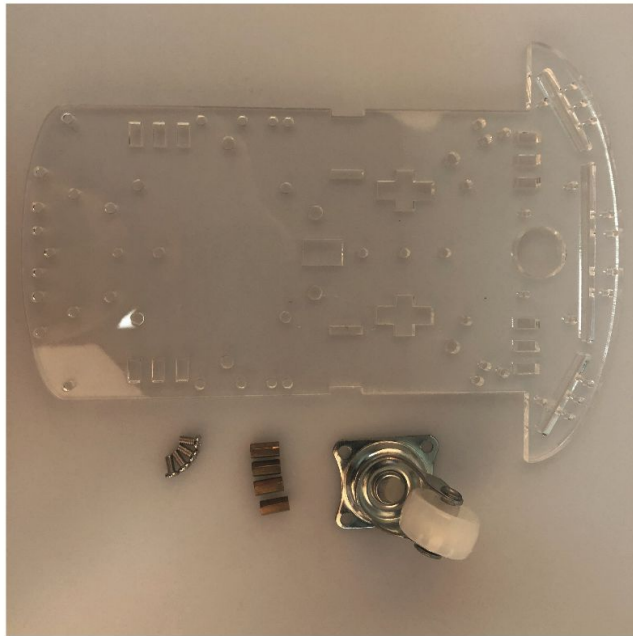
Dagens prosjekt: Selvkjørende robot

- Trenger:



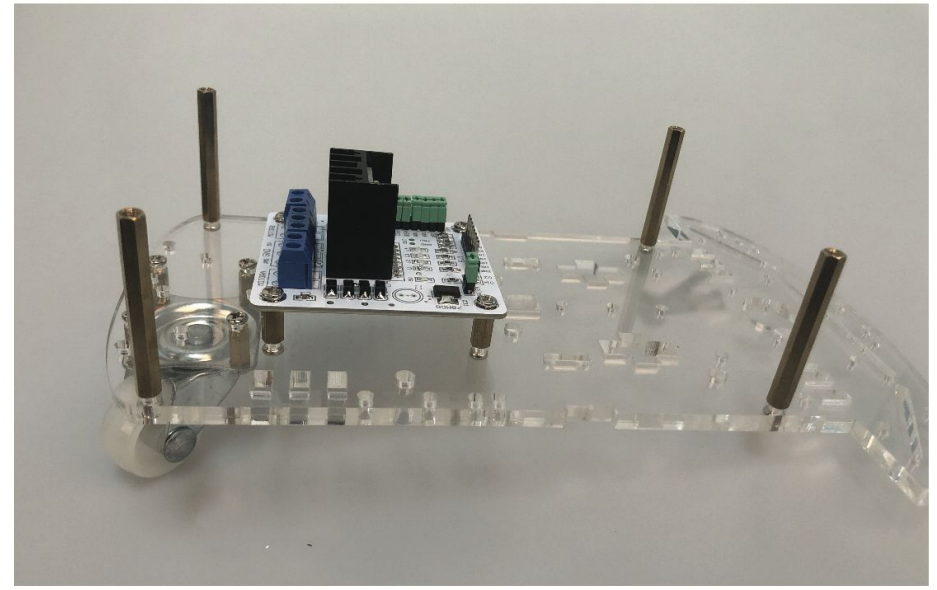
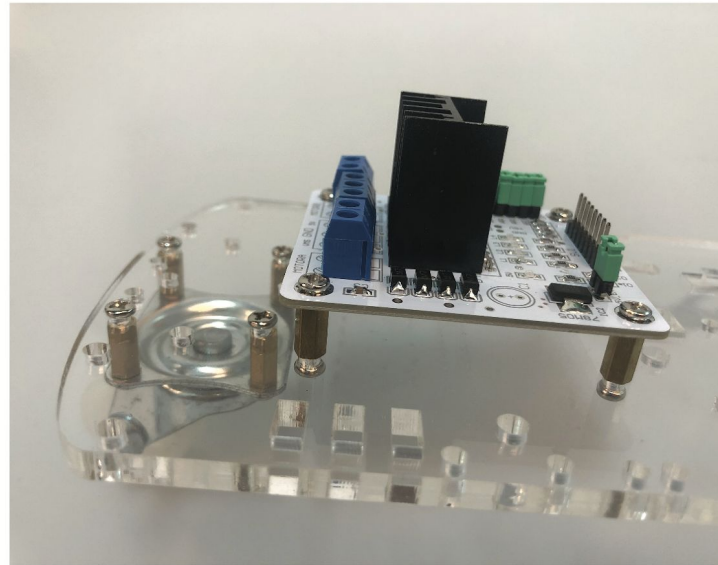
- | | |
|-------------------------|--|
| 1.Arduino Sensor shield | 9.Pleksiglass-chassis (2pcs) |
| 2.Playknowlogy Uno R3 | 10.USB-kabel |
| 3.L298N Motor modul | 11.SG90 Servo |
| 4.Ultralyds-sensor | 12.Strømbryter (Optional) |
| 5.Gummihjul (2pcs) | 13.6 x AA batteriholder |
| 6.DC Motorer (2pcs) | 14. 20 cm hona-hona koblingskabler (20 st) |
| 7.Sensor holder | |
| 8.Støttehjul i metall | |

Fremgangsmåte



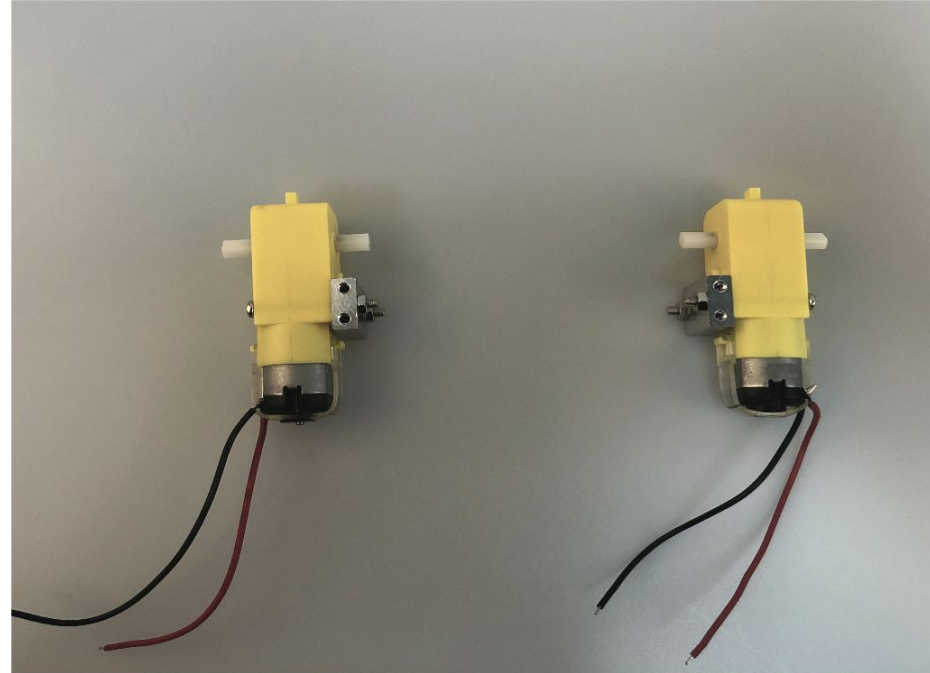
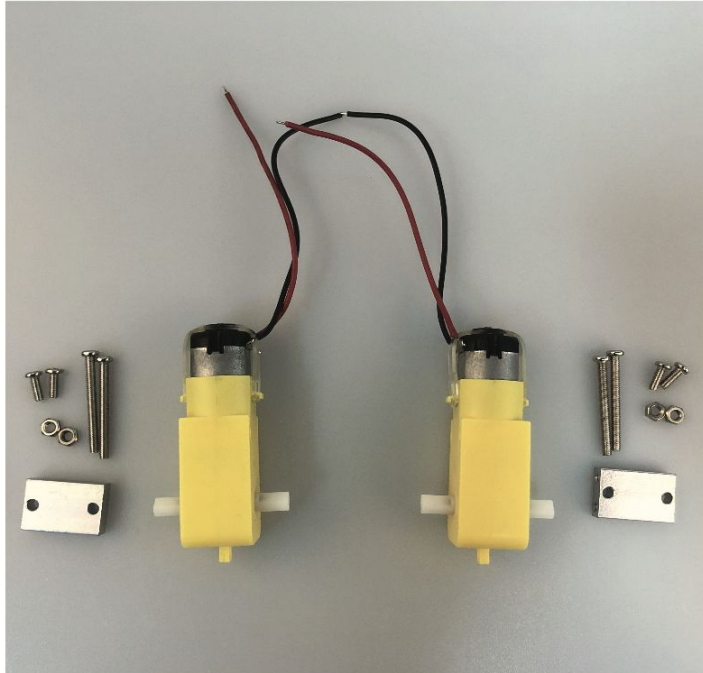
Fjern beskyttelsen fra plexiglass-chassiset. Skru i 4x 6 mm skruer og 4x 10 mm gjenger på støttehjulet. Fest hjulet i chassiset med 4x 8 mm-skruer.

Fremgangsmåte



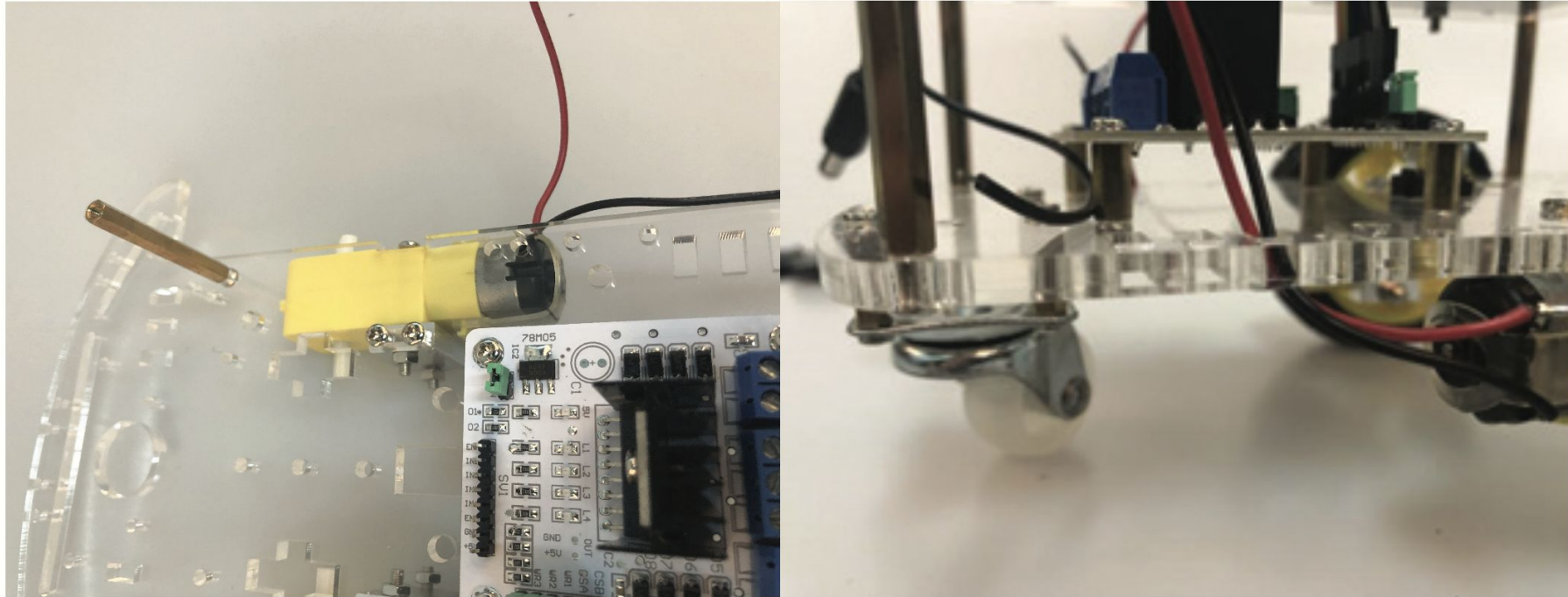
Skru fast L289N-motormodulen i kabinettet med 8x 8 mm-skruer og 4x 10 mm-skruer slik som på bildet ovenfor til høyre. Finn frem 4x 10 mm-skruer og 4x 35 mm-gjenger. Fest dem i kabinettet som vist på bildet ovenfor.

Fremgangsmåte



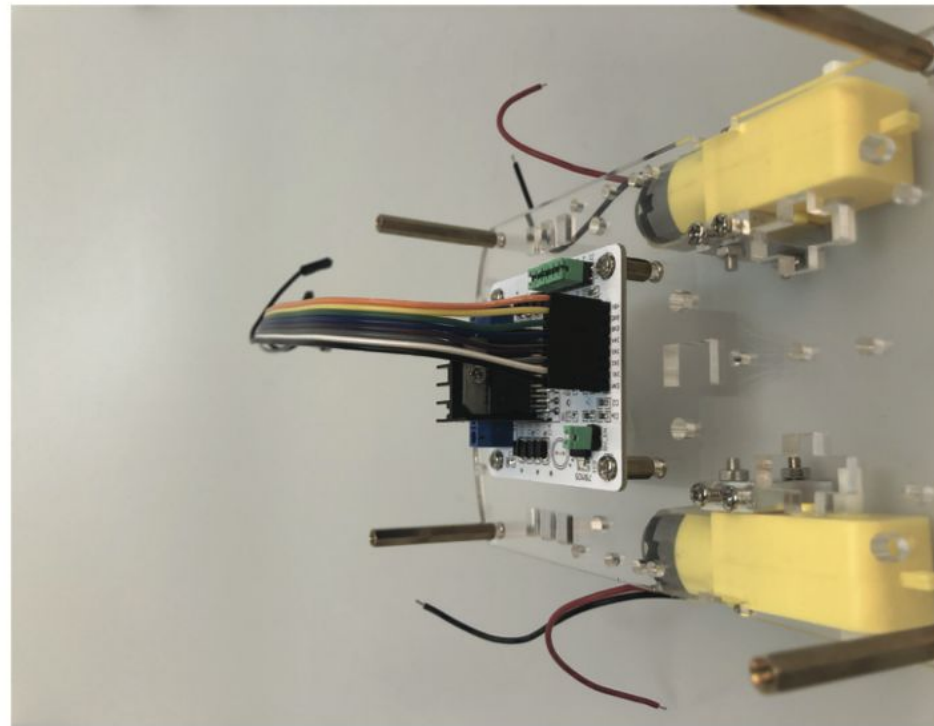
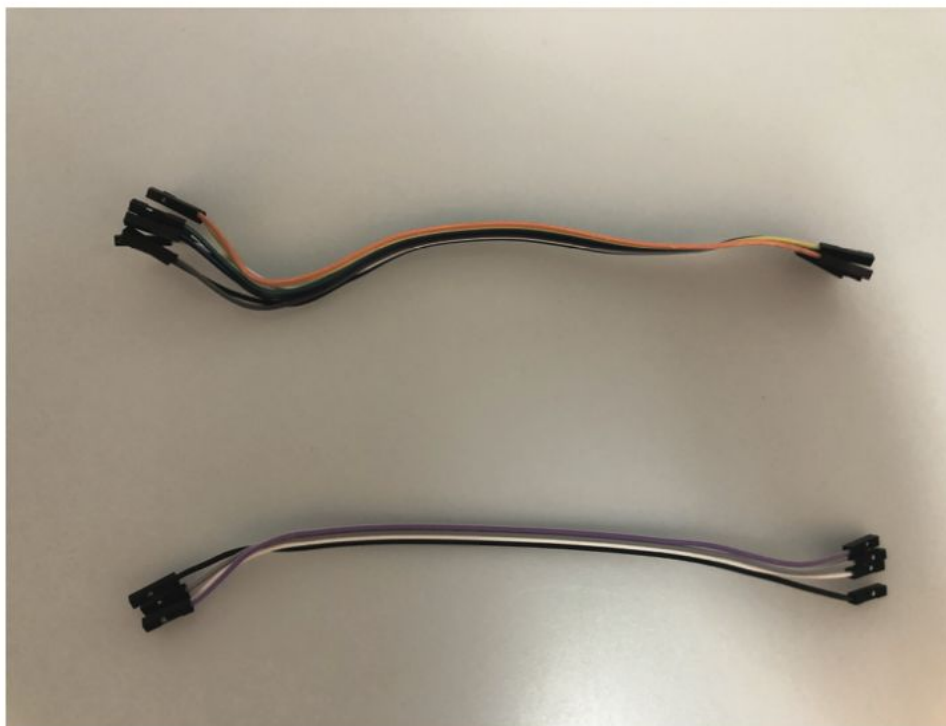
Finn frem delene ovenfor: De to motorene og deres holdere, 4x 35 mm-skruer, 4x 8 mm-skruer og 4x muttere. Plasser holderne på hver sin side av motorene, og fest skruene som vist på bildet.

Fremgangsmåte



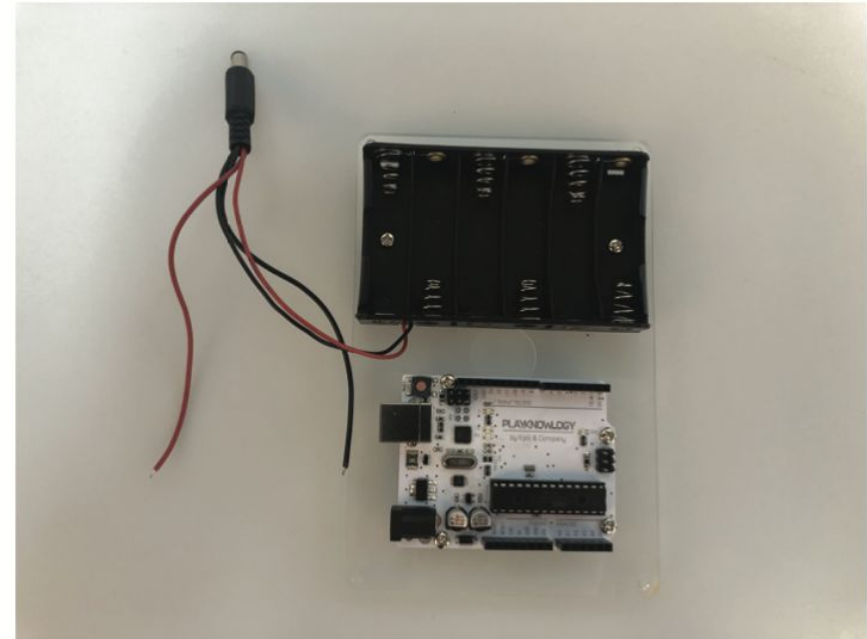
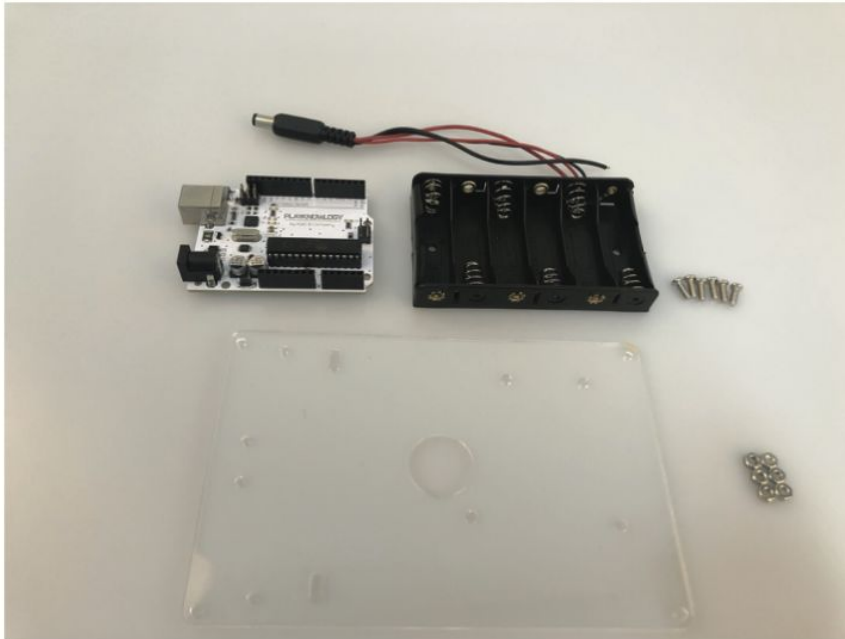
Fest holderne som vist ovenfor. Sikre dem med muttere for å klemme motorene på plass. Resultatet bør se ut som på bildet ovenfor. Fest hjulet til de ytre akslene når du er klar, og dra kablene gjennom hullene slik at de kan nå kortet.

Fremgangsmåte



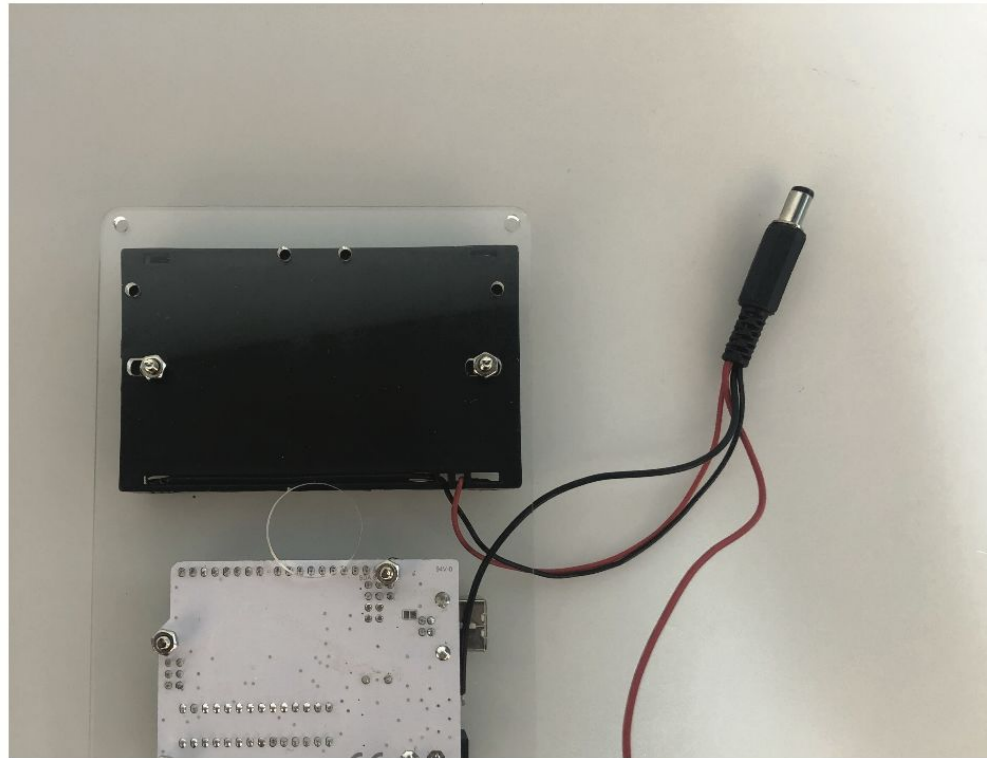
Separer koblingskablene slik at den ene halvdelen har 8 ledere, og den andre 4.

Fremgangsmåte



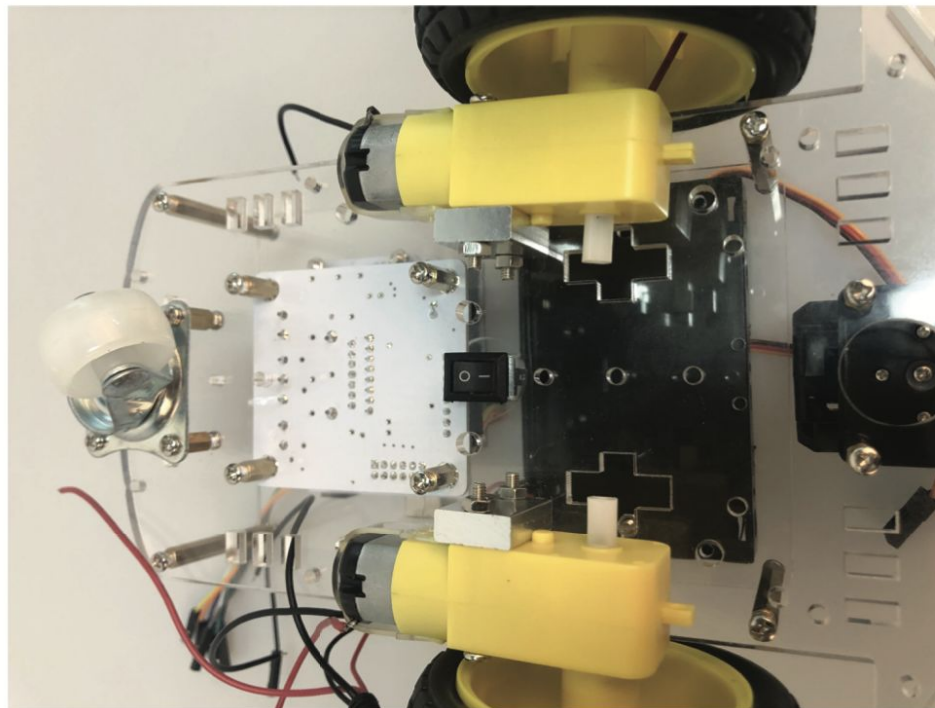
Koble den ene enden av koblingskablene som på bildet ovenfor i motormodulen. La den andre ligge løst (ikke tilkoblet) foreløpig.

Fremgangsmåte



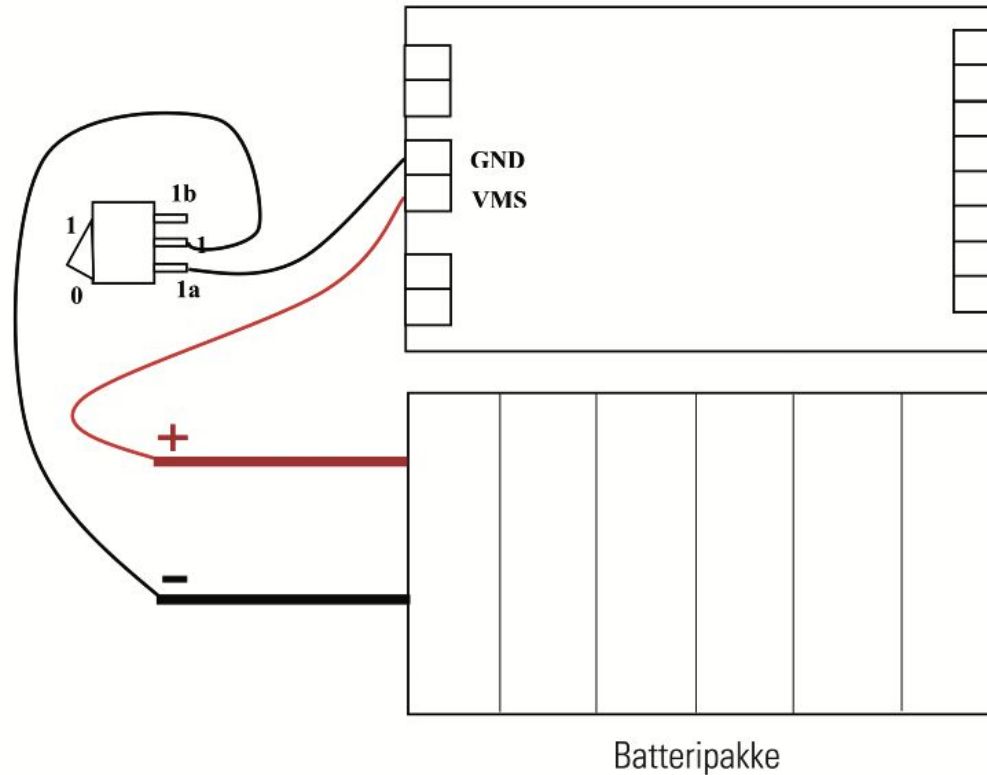
Plasser batteriholderen og UNO-en som på bildene, og fest dem med 6x 10 mm-skruer fra den andre siden. Dra koblingskablene gjennom hullet på chassiset, og koble dem til UNO-kortet.

Fremgangsmåte



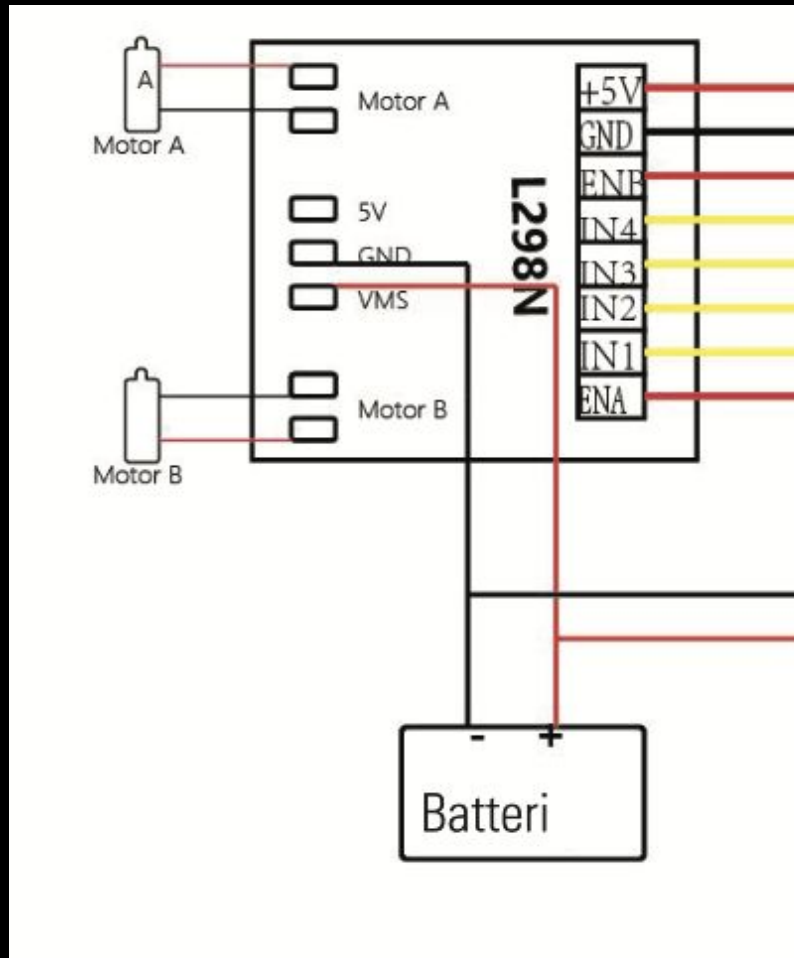
Valgfritt: Du kan koble til en strømbryter for å spare energi. Koble bryteren som på bildet, og lodd strømka-
blene fast i bryteren. Loddeverktøy og tilbehør kreves (følger ikke med).

Fremgangsmåte



Diagrammet ovenfor viser hvordan en strømbryter skal kobles til batteripakken og L289-modulen. Merk: IKKE ha Iduino koblet til datamaskinen samtidig som L289-modulen, siden den kan dra veldig mye strøm fra USB-porten fra datamaskinen og svi den av. Koble ALLTID batteripakken fra enheten før du kobler den til via USB.

Fremgangsmåte



Fremgangsmåte

Ultralydsensoren fungerer som bilens øyne. I følgende trinn skal vi gå gjennom hvordan sensoren og servoen installeres.



Ovenfor vises de komponentene som kreves for disse trinnene.

Fremgangsmåte



Monter FPV-holderen og servoen.



Bruk to borende skruer for å feste FPV-holderen.



Følgende deler trengs for montering. Servo-hornet skal passe inni holderen og må kanskje derfor beskjæres. Bruk en avbitertang eller et annet verktøy for å klippe kantene. Vær forsiktig når du bruker skarpe verktøy.

Fremgangsmåte



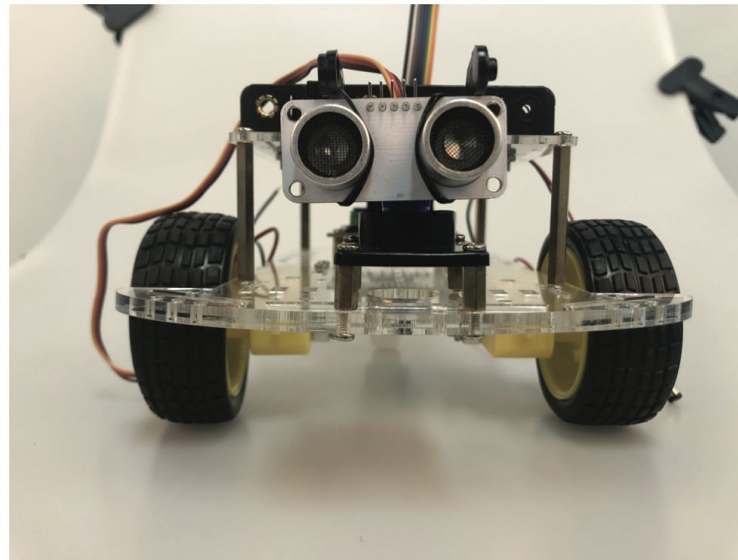
Fest servohornet til holderen med borende-skruer til det sitter godt. Bildet ovenfor viser hvordan det kan se ut. Deretter festes ultralydsensoren til holderen. Bruk buntbånd/strips, snor eller tråd (ingen av delene følger med). Resultatet bør se ut som på bildet nedenfor.



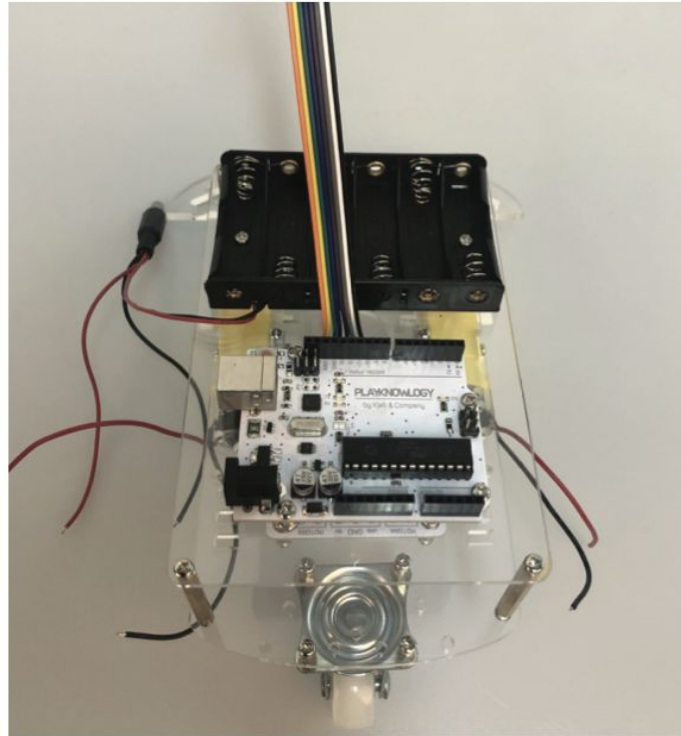
Fremgangsmåte



Skru inn en borende skrue for å feste servoen til sensoren. Fest den som på bildet ovenfor.

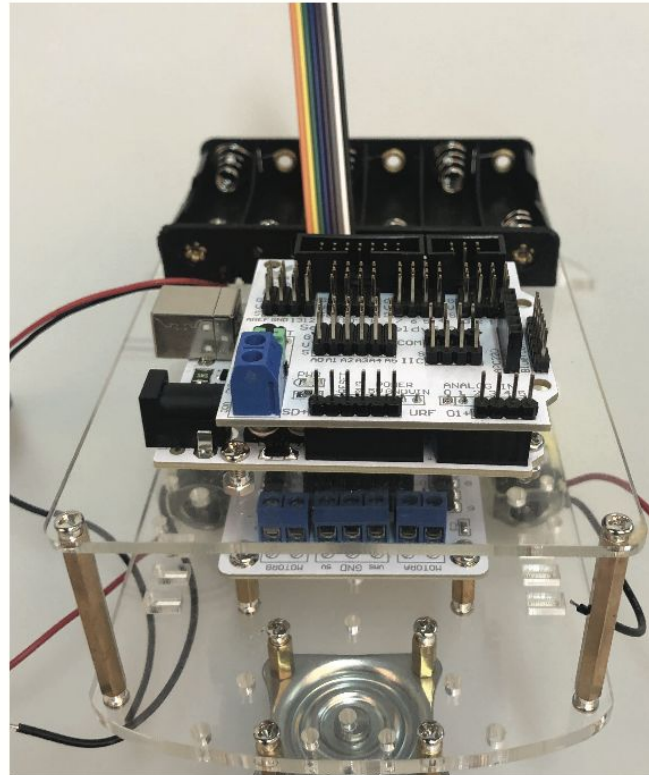


Fremgangsmåte



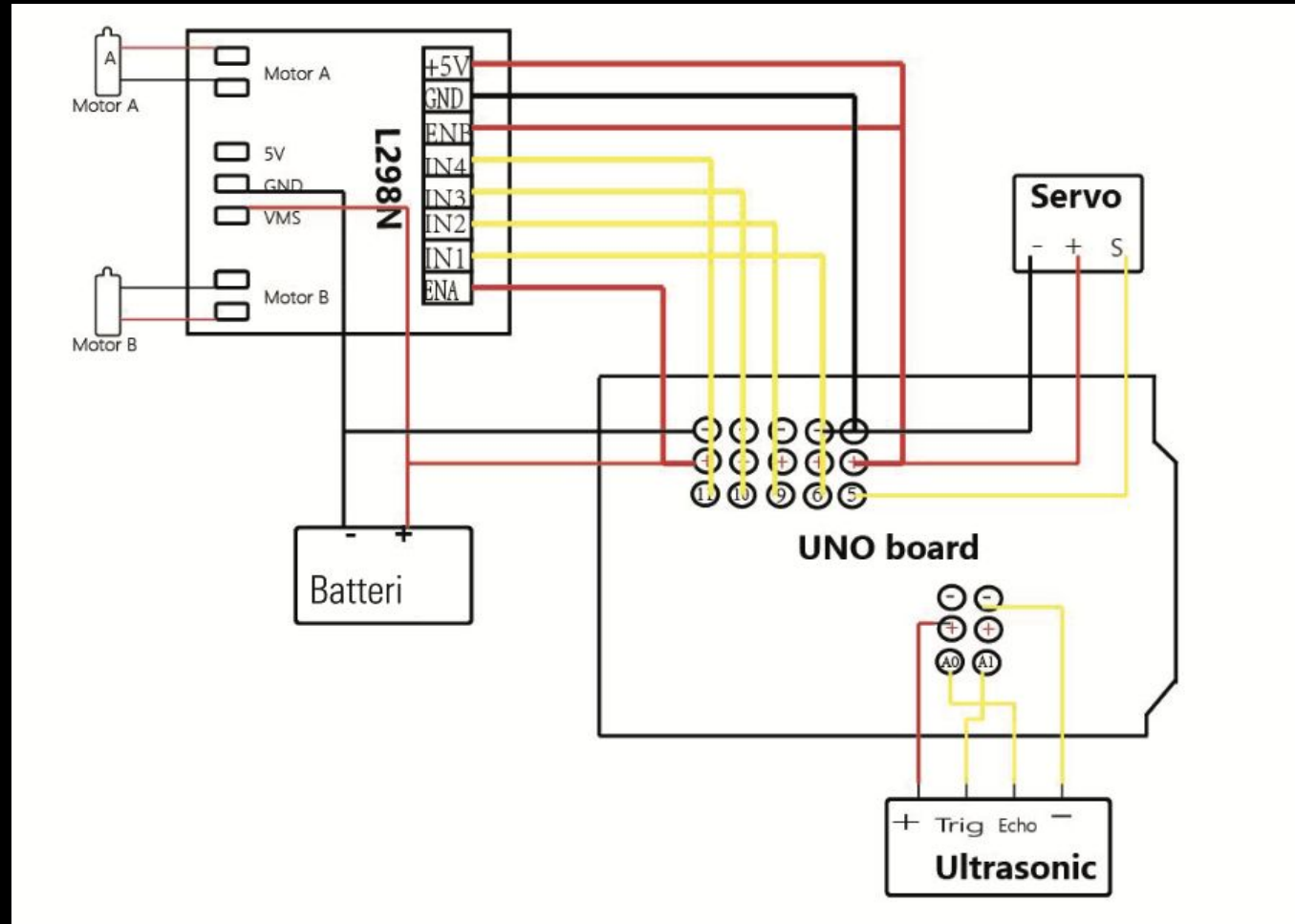
Monter fast toppen av pleksiglass-chassiset med bunnen, og skru dem sammen med 4x 10 mm-skruer. Sluttresultatet bør se ut som på bildet ovenfor.

Fremgangsmåte



Plasser sensor-shieldet på toppen av UNO-kortet, og kontroller at pinnene er justert i forhold til hverandre.

Koblingsskjema



Koden

```
1  #include <Servo.h>
2
3
4  int pinLB=6; // define pin6 as left back connect with IN1
5  int pinLF=9; // define pin9 as left forward connect with IN2
6  int pinRB=10; // define pin10 as right back connect with IN3
7  int pinRF=11; // define pin11 as right forward connect with IN4
8  int inputPin = A0; // define ultrasonic receive pin (Echo)
9  int outputPin =A1; // define ultrasonic send pin(Trig)
10 int Forward_dist = 0; // initial forward distance
11 int Right_dist = 0; // initial right distance
12 int Left_dist = 0; // initial left distance
13 int directionn = 0;
14 Servo myservo; // new myservo
15 int delay_time = 250; // set stable time
16 int Fgo = 8; // forward
17 int Rgo = 6; // turn right
18 int Lgo = 4; // turn left
19 int Bgo = 2; // back
20
```

Koden

```
21
22 void setup()
23 {
24     pinMode(pinLB, OUTPUT);
25     pinMode(pinLF, OUTPUT);
26     pinMode(pinRB, OUTPUT);
27     pinMode(pinRF, OUTPUT);
28     pinMode(inputPin, INPUT);
29     pinMode(outputPin, OUTPUT);
30     myservo.attach(5); // define the servo pin(PWM)
31 }
32
33
34 void advance(int a) // forward
35 {
36     digitalWrite(pinRB, LOW);
37     digitalWrite(pinRF, HIGH);
38     digitalWrite(pinLB, LOW);
39     digitalWrite(pinLF, HIGH);
40     delay(a * 15);
41 }
42
```

```
43
44 void turnR(int d) //turn right
45 {
46     digitalWrite(pinRB, HIGH);
47     digitalWrite(pinRF, LOW);
48     digitalWrite(pinLB, LOW);
49     digitalWrite(pinLF, HIGH);
50     delay(d * 50);
51 }
52
53
54 void turnL(int e) //turn left
55 {
56     digitalWrite(pinRB, LOW);
57     digitalWrite(pinRF, HIGH);
58     digitalWrite(pinLB, HIGH);
59     digitalWrite(pinLF, LOW);
60     delay(e * 50);
61 }
62
```

```
64 void stopp(int f) //stop
65 {
66     digitalWrite(pinRB, LOW);
67     digitalWrite(pinRF, LOW);
68     digitalWrite(pinLB, LOW);
69     digitalWrite(pinLF, LOW);
70     delay(f * 100);
71 }
72
73
74 void back(int g) //back
75 {
76     digitalWrite(pinRB, HIGH);
77     digitalWrite(pinRF, LOW);
78     digitalWrite(pinLB, HIGH);
79     digitalWrite(pinLF, LOW);
80     delay(g * 300);
81 }
```

Koden

```
84 void detection() //test the distance of different direction
85 {
86     int delay_time = 250;
87     delay(200);
88     ask_pin_F(); // read forward distance
89     if(Forward_dist < 10) // if forward distance less than 10
90     {
91         stopp(1);
92         back(2);
93     }
94     if(Forward_dist < 25) // if forward distance less than 25
95     {
96         stopp(1);
97         ask_pin_L();
98         delay(delay_time);
99         ask_pin_R();
100        delay(delay_time);
101        if(Left_dist > Right_dist) // if left distance more than right distance
102        {
103            directionn = Lgo;
104        }
105        if(Left_dist <= Right_dist) // if left distance is less or equal to right distance
106        {
107            directionn = Rgo;
108        }
109
110        directionn = Bgo;
111    }
112    else
113    {
114        directionn = Fgo;
115    }
116 }
117
```

Koden

```
118
119 void ask_pin_F() // test forward distance
120 {
121     myservo.write(90);
122     digitalWrite(outputPin, LOW);
123     delayMicroseconds(2);
124     digitalWrite(outputPin, HIGH);
125     delayMicroseconds(10);
126     digitalWrite(outputPin, LOW);
127     float Fdistance = pulseIn(inputPin, HIGH);
128     Fdistance= Fdistance/58.0;
129     Forward_dist = Fdistance;
130 }
131
132
133 void ask_pin_L() // test left distance
134 {
135     myservo.write(150);
136     delay(delay_time);
137     digitalWrite(outputPin, LOW);
138     delayMicroseconds(2);
139     digitalWrite(outputPin, HIGH);
140     delayMicroseconds(10);
141     digitalWrite(outputPin, LOW);
142     float Ldistance = pulseIn(inputPin, HIGH);
143     Ldistance= Ldistance/58.0;
144     Left_dist = Ldistance;
145 }
146
```

```
147
148 void ask_pin_R() // test right distance
149 {
150     myservo.write(20);
151     delay(delay_time);
152     digitalWrite(outputPin, LOW);
153     delayMicroseconds(2);
154     digitalWrite(outputPin, HIGH);
155     delayMicroseconds(10);
156     digitalWrite(outputPin, LOW);
157     float Rdistance = pulseIn(inputPin, HIGH);
158     Rdistance= Rdistance/58.0;
159     Right_dist = Rdistance;
160 }
161
```

Koden

```
162  
163 void loop()  
164 {  
165     myservo.write(90);  
166     detection();  
167     if(directionn == 2)  
168     {  
169         back(3);  
170         turnL(2);  
171     }  
172     if(directionn == 6)  
173     {  
174         back(1);  
175         turnR(6);  
176     }  
177     if(directionn == 4)  
178     {  
179         back(1);  
180         turnL(6);  
181     }  
182     if(directionn == 8)  
183     {  
184         advance(1);  
185     }  
186 }
```


Spørsmål?
