

FølgerVenstre

```
// Eksempel med master og slave kommunikasjon via I2C
```

```
// Slave har adresse fra 2 til 127 (her 2)
```

```
// VENSTRE
```

```
// I2C biblioteket
```

```
#include <Wire.h>
```

```
#include <Servo.h> //Ro Høyreside
```

```
#include <Servo.h> //Ro Høyreside
```

```
Servo servoRo;
```

```
Servo servoHev;
```

```
// Bruk av timer 1 til å styre hendelser i loop()
```

```
// timer1 vil gi 100 avbrudd i sekundet
```

```
bool b_Second = false; // Settes i timer 1 avbrudd, brukes i loop()
```

```
bool b_HundredthOfSecond = false; // Settes i timer 1 avbrudd, brukes i loop()
```

```
bool w_Fremover = false; // Brukes i loop for å kjøre båten frem.
```

```
long _HundredthOfSecond = 0; // Økes med en i timer1 avbrudd (gir overflyt  
etter ca 8 måneder)
```

```
bool a_Venstre = false; //brukes i loop
```

```
bool d_Hoyre = false;
```

```
bool s_Bakover = false;
```

```
bool p_Party = false;
```

```
bool stopp = false;
```

```
bool brems = false;
```

```
int LEDR = 13;
```

```
int LEDG = 12;
```

```
bool H_G_Blink = false;
```

```
bool V_R_Blink = false;
```

```
int c; // signal fra Webserver og Master
```

```
int x=0; //speed ting
```

```
char sx[3];
```

```
int vinkelRo1 = 170; //bak
```

```
int vinkelRo2 = 70; //frem
```

```
int vinkelHev1 = 63; //opp
```

```
int vinkelHev2 = 25; //ned
```

```
//hastighet
```

```
int speed=600;
```

```
// Kode som kjøres kun en gang.
```

```
void setup() {
```

```
  Wire.begin(2);
```

```
// "meld inn" på I2C bussen med adresse 2
```

```
  Wire.onReceive(receive_I2C_Event); // registrer hendelse
```

```
  Serial.begin(9600);
```

FølgerVenstre

```
servoRo.attach(10);
servoHev.attach(9);
pinMode(LED_R, OUTPUT);
pinMode(LED_G, OUTPUT);

servoHev.write(63); // Utgangsposisjon
servoRo.write(15); // Utgangsposisjon

}

void loop() {

  digitalWrite(LED_R, HIGH); // LYS
  digitalWrite(LED_G, HIGH);

  if(w_Fremover)
  {

    servoHev.write(vinkelHev1);
    delay(speed);
    servoRo.write(vinkelRo2); // Går fremover
    delay(speed);

    do {
      w_Fremover=false;
      Serial.println("Båten går fremover en gang");
      servoHev.write(vinkelHev2);
      delay(speed);
      servoRo.write(vinkelRo1); // hev 1 er opp
      delay(speed);
      servoHev.write(vinkelHev1);
      delay(speed);
      servoRo.write(vinkelRo2); // hev 2 er opp
      delay(speed);
    } while (c == 1 || c == 8 || c == 9);

  }

  if(stopp)
  {
    do {
      stopp=false;
      Serial.println("STOPP");
      servoRo.write(15); // Stopp/nøytral
      delay(speed);
      servoHev.write(vinkelHev1);
      delay(speed);
      servoRo.write(15);
      delay(speed);
    } while (stopp);
  }
}
```

```

                                FølgerVenstre
servoHev.write(vinkelHev1);
delay(speed);
} while (c == 2);
}

if(a_Venstre)
{
    tone(8,1000,250);
    delay(350);                // Tuter 2 ganger
    tone(8,1000,250);

    servoHev.write(vinkelHev1);
    delay(speed);
    servoRo.write(vinkelRo1);    // Barbord/Venstre
    delay(speed);

    do {
        a_Venstre=false;
        servoHev.write(vinkelHev2);
        delay(speed);
        servoRo.write(vinkelRo2);
        delay(speed);
        servoHev.write(vinkelHev1);
        delay(speed);
        servoRo.write(vinkelRo1);
        delay(speed);
    } while (c == 3);
}

if(d_Hoyre )
{
    tone(8,1000,250);          // Tuter en gang

    servoHev.write(vinkelHev1);
    delay(speed);
    servoRo.write(vinkelRo2);    // Styrbord/Høyre
    delay(speed);

    do {
        d_Hoyre=false;
        Serial.println("Båten går fremover en gang");
        servoHev.write(vinkelHev2);
        delay(speed);
        servoRo.write(vinkelRo1);
        delay(speed);
        servoHev.write(vinkelHev1);
        delay(speed);
        servoRo.write(vinkelRo2);
        delay(speed);
    } while (c == 4);
}

```

FølgerVenstre

```
}
```

```
if(brems)
{
  do {
    brems=false;
    Serial.println("bremser");
    servoRo.write(90);
    delay(200);
    servoHev.write(vinkelHev2);           // Bremser
    delay(100);
    servoRo.write(90);
    delay(200);
    servoHev.write(vinkelHev2);
    delay(100);
  } while (c == 5);
}
```

```
if(s_Bakover)
{
  Serial.println("båten går bakover");
  tone(8,1000,250);
  delay(350);
  tone(8,1000,250);           // Tuter 3 ganger
  delay(350);
  tone(8,1000,250);

  servoHev.write(vinkelHev1);
  delay(speed);
  servoRo.write(vinkelRo1);
  delay(speed);

  do {
    s_Bakover=false;
    servoHev.write(vinkelHev2);           // Bakover
    delay(speed);
    servoRo.write(vinkelRo2);
    delay(speed);
    servoHev.write(vinkelHev1);
    delay(speed);
    servoRo.write(vinkelRo1);
    delay(speed);
  }while (c==6);
}
```

```
if(p_Party)
```

FølgerVenstre

```
{
  do {

    p_Party=false;
    Serial.println("SALUTPARTY");
    servoRo.write(90);
    delay(100);
    servoHev.write(vinkelHev1);
    delay(100);
    servoRo.write(90);
    delay(100);
    servoHev.write(vinkelHev2);
    delay(100);
  } while (c == 7);
}
}
```

```
// Funksjonen receive_I2C_Event utføres hver gang master sender data til slaven
// Funksjonen er listet opp som en hendelse (event) i setup()
void receive_I2C_Event(int) {
```

```
while (Wire.available()) { // loop gjennom alle tegn
  c = Wire.read();          // motta tegn som byte
  Serial.println(c);        // print tegnet på serieportenr
  if (c == 1) {
    w_Fremover = true;      // Sender kommando om å kjøre båten fremover
  }
  if (c == 3){
    a_Venstre = true;
  }
  if (c == 4){
    d_Hoyre = true;
  }
  if (c == 6){
    s_Bakover = true;
  }
  if (c== 7){
    p_Party = true;
  }
  if (c== 2){
    stopp = true;
  }
  if (c== 5){
    brems = true;
  }
}
```

```
if(c==8){ //øker hastighet --> minker delay
  speed-=100;
  w_Fremover = true;
```

FølgerVenstre

}

```
if(c==9){ //minker hastighet --> øker delay  
    speed+=100;  
    w_Fremover = true;  
}
```

}

}