```
#include <Zumo32U4.h>
                                        //Inkluderer Zumobibliotek
#include "math.h"
Zumo32U4Motors motors;
                                          //Forenkling av navn slik at det blir enklere å bruke
funksjonene videre i koden.
Zumo32U4LCD lcd;
Zumo32U4Encoders encoders;
unsigned long last_time = 0;
                                         //Variabler som brukes til å regne ut ett sekund, ti sekund og
ett minutt
unsigned long last ten sec = 0;
unsigned long last_minute = 0;
unsigned long one_second = 1000;
unsigned long ten_seconds = 10000;
unsigned long one_minute = 60000;
int left_speed = 400;
int right_speed = 400;
float motorSpeed = (left_speed + right_speed) / 2;
float batteryLevel = 100;
float meters = 0;
                                    //Total avstand kjørt siden start av zumo
float meters_per_second = 0;
                                           //Gj.snitthastigeten det siste sekundet
float average_speed = 0;
                                        //Gj.snitthastigeten det siste minuttet
float meters_per_minute = 0;
                                           //Antall meter kjørt det siste minuttet
float maxSpeed = 0;
                                      //Maks hastighet det siste minuttet
```

```
void setup()
Serial.begin(9600);
motors.setSpeeds(left_speed, right_speed);
                                                  //Zumo får fart i den hastigeten som står i
left_speed og right_speed over
}
void loop()
{
 delay(10);
if((millis()-last_time) >= one_second)
                                             //If-funksjon som går inn hvert sekund
{
  int leftCount = encoders.getCountsAndResetLeft(); //Variabler som brukes til å beregne zumo har
kjørt
  int rightCount = encoders.getCountsAndResetRight();
  float averageCount = (leftCount+rightCount)/2;
  float this meter = averageCount/8000.0;
                                                 //Konvertere fra motorcounts til meter
                                         //Teller antall meter kjørt
  meters += this meter;
  unsigned long myTime = millis()-last time;
                                                 //Får tiden når programmet går inn i if funksjonen
  meters per second = (this meter/myTime)*1000;
                                                       //Regner ut meter per sekund
  if(maxSpeed < meters_per_second) maxSpeed = meters_per_second; //Lagrer maks hastighet til en
variabel
 lcd.clear();
                                  //Viser avstand og hastighet på zumoens lcdskjerm
 lcd.gotoXY(0,0);
 lcd.print(String(meters) + "m");
 lcd.gotoXY(0,1);
 lcd.print(String(meters per second) + "m/s");
```

```
last_time = millis();
                                     //Oppdaterer variabel last_time
}
 if((millis()-last_ten_sec) >= ten_seconds)
                                               //If-funksjon som går inn hvert tiende sekund
  batteryLevel = batteryCalc(batteryLevel, motorSpeed); //Bruker batteryCalc-funksjonen til å regne ut
batterinivået
  EEPROM.write(1, batteryLevel);
  lcd.clear();
                                   //Viser batterinivået på skjermen
  lcd.gotoXY(0,0);
  lcd.print(String(batteryLevel) + "%");
  delay(500);
                                        //Oppdaterer last_ten_sec
 last_ten_sec = millis();
 }
if((millis()-last_minute) >= one_minute)
                                        //If-funksjon som går inn hvert minutt
  unsigned long my_time = millis()-last_minute;
  meters_per_minute = meters;
  average_speed = (meters/my_time)*1000;
                                                    //Regner ut gjennomsnittshastighet i m/s
  average_speed = (meters_per_minute/my_time)*1000;
                                                           //Regner ut gjennomsnittshastighet i m/s
  lcd.clear();
  lcd.gotoXY(0,0);
```

```
lcd.print("M:" + String(maxSpeed) + "m/s");
                                                    //Printer ut maks hastighet hvert minutt
  lcd.gotoXY(0,1);
 lcd.print("A:" + String(average_speed) + "m/s");
                                                    //Printer ut gjennomsnittshastighet hvert minutt
  delay(1000);
  meters_per_minute = 0;
                                             //Resetter variabler
  maxSpeed = 0;
  last_minute = millis();
                                          //Oppdaterer last_minute
  }
 if (batteryLevel > 30)
                                         //Kode som bruker zumoens LED-lys til å indikere batterinivå
  {
  ledYellow(0);
  ledRed(0);
  ledGreen(1);
  }
else if (batteryLevel > 20)
{
ledYellow(1);
ledRed(0);
ledGreen(0);
}
else if (batteryLevel > 10)
{
ledYellow(0);
ledRed(1);
ledGreen(0);
```

```
}
else if (batteryLevel == 0)
 {
  motors.setSpeeds(0,0);
                                              //Stopper bilen når batteriet er tomt
ledYellow(0);
ledRed(0);
ledGreen(0);
 }
}
int batteryCalc(float batteryLevel, float motorSpeed)
                                                        //Funksjon som regner ut batterinivå som
funkson av hastighet og distanse kjørt
{
                                               //Hvor mye batteri zumo bruker per ti sekunder ved full
 const int batteryDrain = 10;
hastighet
 const int topSpeed = 400;
 float actBatteryDrain = (motorSpeed / topSpeed)* batteryDrain; //Faktisk batteritap avhengig av
farten til zumo
 if (fmod(meters, 1.0) >= 0.0 \&\& fmod(meters, 1.0) < 0.5) //Bruker avstand kjørt til sjekke om
batteryLevel skal oppdateres
  if (batteryLevel > 0)
  {
   batteryLevel -= actBatteryDrain;
                                                  //Trekker batteriforbruk de siste ti sekundene
   if (batteryLevel <= 0)
                                            //Hvis batterinivået blir mindre enn null, blir det satt til null
   {
    batteryLevel = 0;
```

```
}
}
return batteryLevel; //Returnerer utregnet batterinivå
}
```