

# Komponent Reference

Af: Max Bursell og Michael Hansen, Din Camp 2023, version 0.94

Mekanisk, fotografisk, elektronisk eller anden gengivelse af denne bog eller dele heraf er kun tilladt i henhold til Copydan regler.

[www.dincamp.dk](http://www.dincamp.dk)



## Inventorslinjen

Dette dokument beskriver hvordan de forskellige komponenter bruges, som du finder i vores komponent samling. Referencen indeholder både en beskrivelse af hvordan hardwaren skal forbindes, hvilket Arduino bibliotek som skal installeres og et software eksempel på, hvordan komponenterne tilgås. Nogle udvalgte komponenter har en mere omfattende forklaring end andre, da de enten bruges oftere eller er mere komplekse at anvende.

Til sidst i dokumentet er der en C/C++ programmeringsguide samt en arduino guide.

Har du brug for hjælp til Arduino funktionerne, kan du benytte dette link:

<https://www.arduino.cc/reference/en/>

## Indholdsfortegnelse

1	LED .....	3	21	Potentiometeret .....	34
2	Neopixel.....	4	22	Afstandssensor.....	35
3	LCD 16x2 display.....	6	23	Linjesensor (infrarød).....	37
4	7-segment display TM1637 .....	9	24	Farvesensor.....	38
5	LED matrix 8x8 single.....	10	25	Temperatursensor DHT11 .....	40
6	LED matrix 8x8 chain .....	11	26	Lydsensor .....	42
7	OLED 0.96" display.....	13	27	MP3 afspiller .....	43
8	Buzzer .....	15	28	Gyro .....	45
9	Relæ kontrol .....	16	29	RFID .....	47
10	Knap, kontakt.....	17	30	RTC .....	48
11	Keypad 4x4 .....	19	31	IR remote .....	51
12	Joystick.....	22	32	Bluetooth .....	55
13	PIR motion sensor.....	24	33	C/C++ Programmeringsguide .....	58
14	Tilt/ryste sensor.....	24	34	Arduino bibliotek .....	59
15	Lysfølsom modstand.....	25	35	Arduino Uno vs Arduino Nano .....	60
16	Waterlevel sensor.....	26	36	Bluetooth parring.....	61
17	Rotary encoder .....	28	37	Formater SD kort .....	64
18	Motor med L293D .....	30	38	Bruges til kopi .....	65
19	Servo .....	31			
20	Stepmotor.....	32			

# 1 LED

Få en LED til at blinke. LED'en findes i flere farver.  
Der skal bruges en modstand på  $220\ \Omega$ .

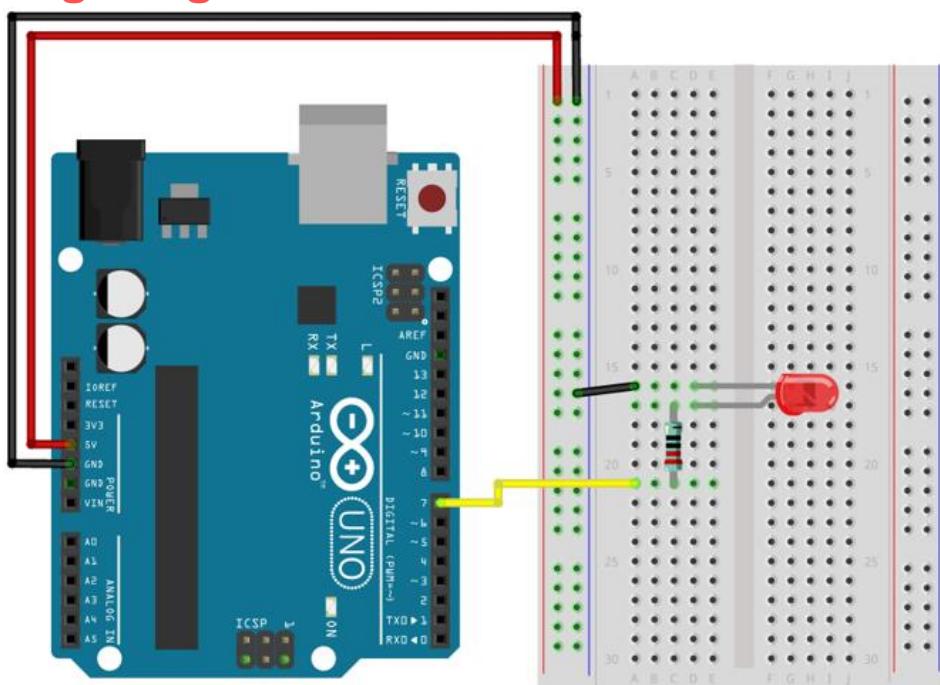


Modstand  $220\ \Omega$



To røde streger.

## 1.1 Ledningsdiagrammer



## 1.2 Software eksempel

```
#define LED_PIN 7

void setup() {
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_PIN, HIGH);
    delay(500);
    digitalWrite(LED_PIN, LOW);
    delay(500);
}
```

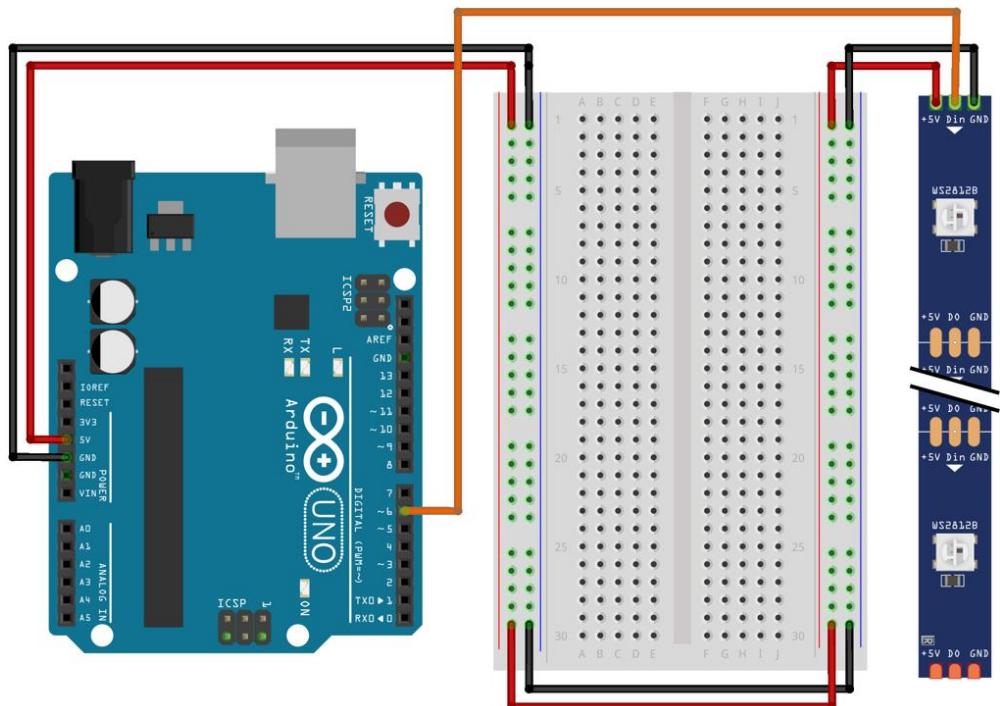
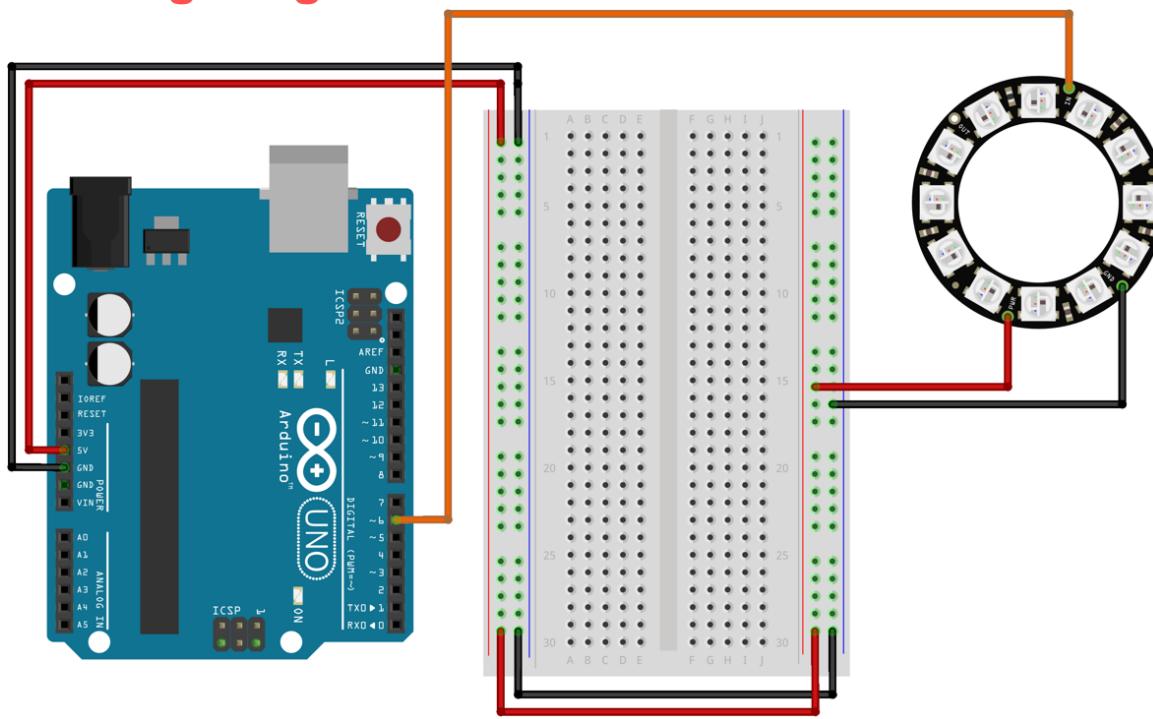
# 2 Neopixel

En neopixel er en 'LED' som kan lyse i 'alle' farver. De kan sidde flere sammen i en lang kæde og er ret simple at få til at virke. Vi har flere forskellige typer af dem, som alle virker ens, men ser forskellige ud. Der er derfor to forskellige ledningsdiagrammer, men koden er ens.

Husk Arduino'en kan **max trække 12 neopixels**.



## 2.1 Ledningsdiagrammer



## 2.2 Arduino bibliotek

Følgende bibliotek skal installeres: *FastLED by Daniel Garcia*.

[FastLED](#)

by Daniel Garcia Version 3.5.0 **INSTALLED**

**Multi-platform library for controlling dozens of different types of LEDs along with optimized math, effect, and noise functions.**

Multi-platform library for controlling dozens of different types of LEDs along with optimized math, effect, and noise functions.

[More info](#)

## 2.3 Software eksempel, individuel styring



```
#include <FastLED.h>

#define NUM_LEDS 8 // Din arduino kan højest levere strøm til 12 LEDer!
#define LED_PIN 6

CRGB leds[NUM_LEDS];

int red;
int green;
int blue;

void setup() {
  FastLED.addLeds<WS2812B, LED_PIN, GRB>(leds, NUM_LEDS);
  FastLED.setBrightness(50); // Gå ikke over 100!
}

void loop() {
  red = 255;
  green = 0;
  blue = 0;
  FastLED.clear();
  leds[0] = CRGB(red, green, blue);
  leds[7] = CRGB(red, green, blue);
  FastLED.show();
  delay(500);

  red = 0;
  green = 0;
  blue = 255;
  FastLED.clear();
  leds[3] = CRGB(red, green, blue);
  leds[4] = CRGB(red, green, blue);
  FastLED.show();
  delay(500);
}
```

## 2.4 Software eksempel, Colour Fade

0 ----- 127 ----- 255



HUE

```
#include <FastLED.h>

#define NUM_LEDS 8 // Din arduino kan højest levele strøm til 12 LEDer!
#define LED_PIN 6

CRGB leds[NUM_LEDS];

byte lightLevel = 255;           // 0 - 255
byte hueValue;

int fadeSpeed = 10;
long lastTime;

void setup() {
    FastLED.addLeds<WS2812B, LED_PIN, GRB>(leds, NUM_LEDS);
    FastLED.setBrightness(100);    // Gå ikke over end 100!
    lastTime = millis();
}

void loop() {
    if(millis() > lastTime + fadeSpeed) {
        hueValue++;
        lastTime = millis();
    }

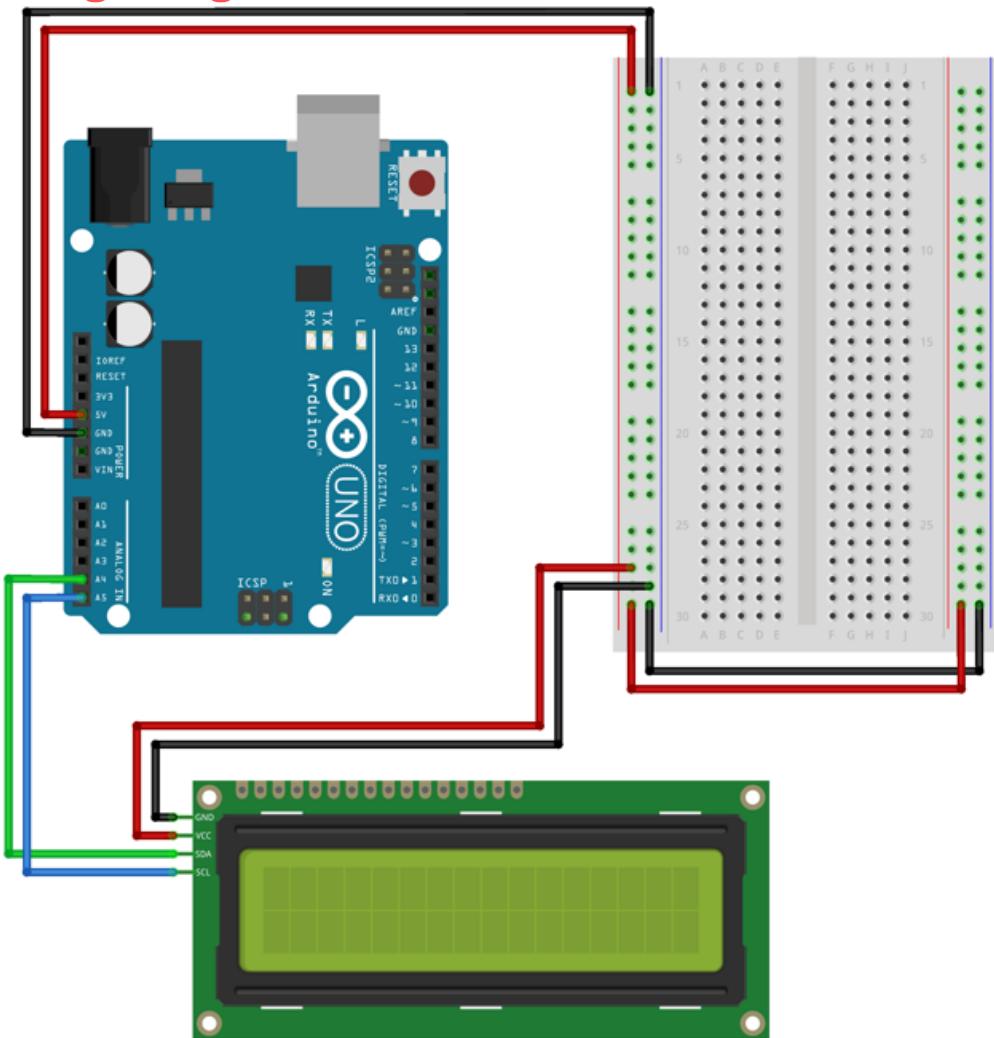
    FastLED.clear();
    for(int i = 0; i < NUM_LEDS; i++) {
        leds[i] = CHSV(hueValue, 255, lightLevel);
    }
    FastLED.show();
}
```

## 3 LCD 16x2 display

LCD display med to linjer, hvor hver linje har plads til 16 bogstaver. Man kan også styre baggrundsbelysningen.



## 3.1 Ledningsdiagrammer



## 3.2 Arduino bibliotek

Følgende bibliotek skal installeres: *LiquidCrystal I2C* by Frank de Brabander.

**LiquidCrystal I2C** by Frank de Brabander

1.1.2 installed

The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES. A library for I2C LCD displays.

[More info](#)

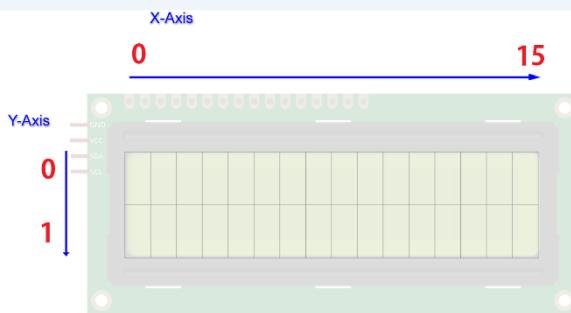
### 3.3 Software eksempel

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MAKERCAMP");
  delay(100);
}
```



### 3.4 Software eksempel, lav dit eget tegn

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

byte happy[] = {
  B00000,
  B11011,
  B11011,
  B00000,
  B10001,
  B10001,
  B11111,
  B00000
} ;

byte sad[] = {
  B00000,
  B11011,
  B11011,
  B00000,
  B00000,
  B11111,
  B10001,
  B10001
} ;

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.createChar(0, happy);
  lcd.createChar(1, sad);
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.write(0);
  delay(1000);
}
```



```

    lcd.setCursor(0, 0);
    lcd.write(1);
    delay(1000);
}

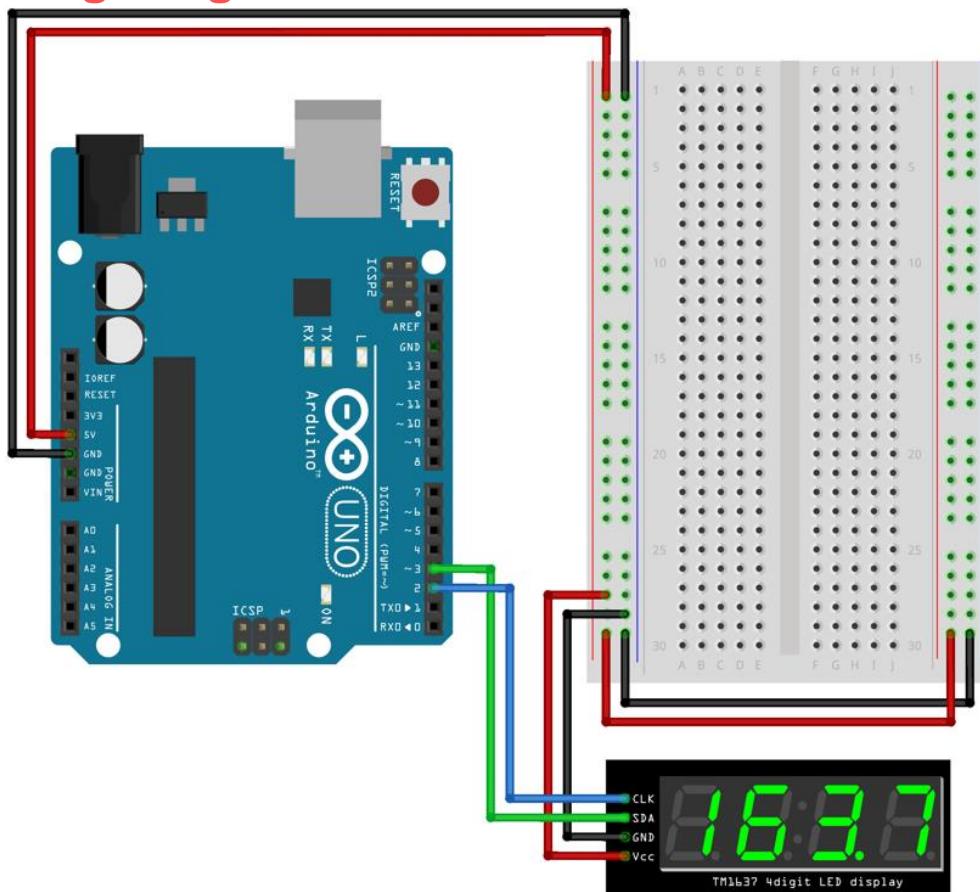
```

## 4 7-segment display TM1637

Display med fire 7-segment displays. Kan bruges til at vise tal.



### 4.1 Ledningsdiagrammer



### 4.2 Arduino bibliotek

Følgende bibliotek skal installeres: *arduino-tm1637-master*

Brug linket for at download biblioteket.

[https://drive.google.com/file/d/1dQ5NRC4Y\\_BeqBaF4IPS-b7xCyRDyAKIS/view?usp=sharing](https://drive.google.com/file/d/1dQ5NRC4Y_BeqBaF4IPS-b7xCyRDyAKIS/view?usp=sharing)

## 4.3 Software eksempel

```
#include "SevenSegmentTM1637.h"

const byte CLK = 4;
const byte DIO = 5;
SevenSegmentTM1637 display(CLK, DIO);

void setup() {
  Serial.begin(9600);
  display.begin();
}

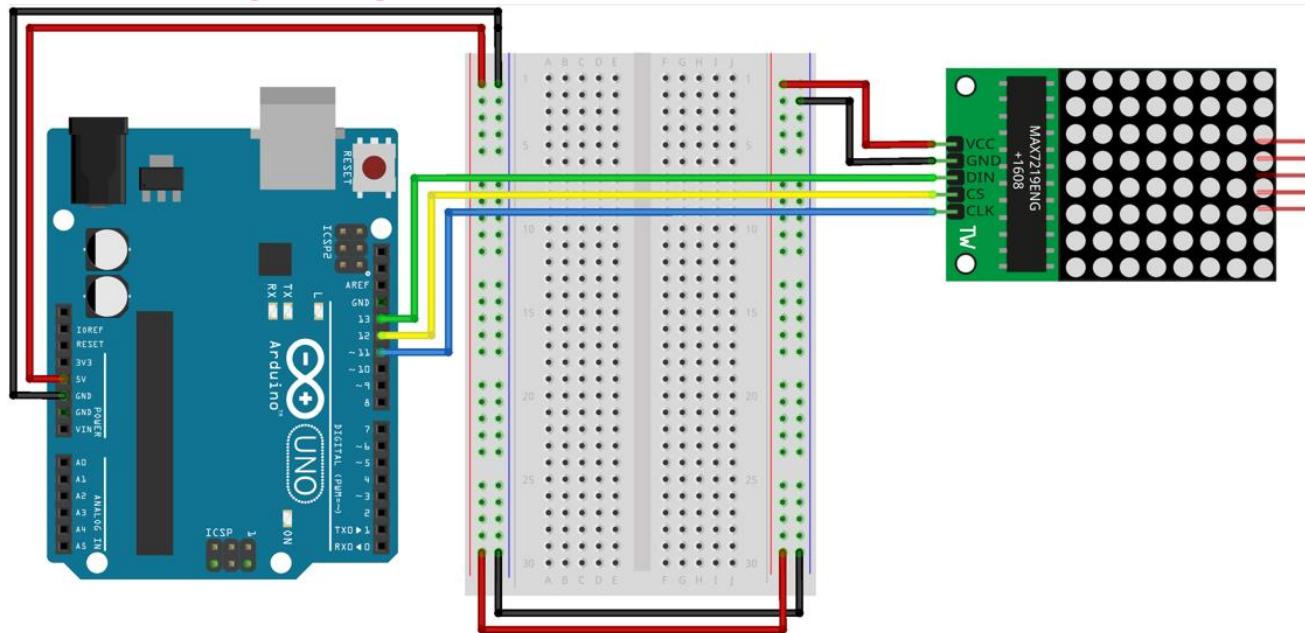
void loop() {
  display.print(1234);
  delay(1000);
  display.print("CAMP");
  delay(1000);
}
```

## 5 LED matrix 8x8 single

LED matrix med 8x8 røde LEDer, hvor hver enkelt LED kan styres.



### 5.1 Ledningsdiagrammer



## 5.2 Arduino bibliotek

Følgende bibliotek skal installeres: *MD\_MAX72XX*.

[MD\\_MAX72XX by majicDesigns](#)

[3.4.1 installed](#)

Allows the programmer to use the LED matrix as a pixel addressable display. Implements functions that allow the MAX72xx (eg, MAX7219) to be used for LED matrices (64 individual LEDs)

[More info](#)

## 5.3 Software eksempel

```
#include <MD_MAX72xx.h>

#define DATA_PIN 13
#define CS_PIN 12
#define CLK_PIN 11

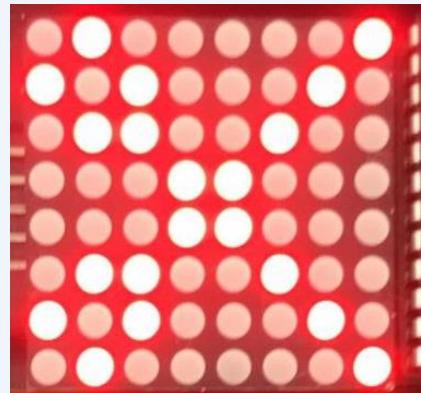
MD_MAX72XX mx = MD_MAX72XX(MD_MAX72XX::PAROLA_HW, DATA_PIN, CLK_PIN, CS_PIN);

void setup() {
    mx.begin();
}

void loop(){
    scissor();
    delay(2000);

    mx.clear();
    delay(2000);
}

void scissor() {
    byte rows[8] = {
        B01000001,
        B10100010,
        B01100100,
        B00011000,
        B00011000,
        B01100100,
        B10100010,
        B01000001,
    };
    for(int i = 0; i <= 7; i++) {
        mx.setRow(0, i, rows[i]);
    }
}
```



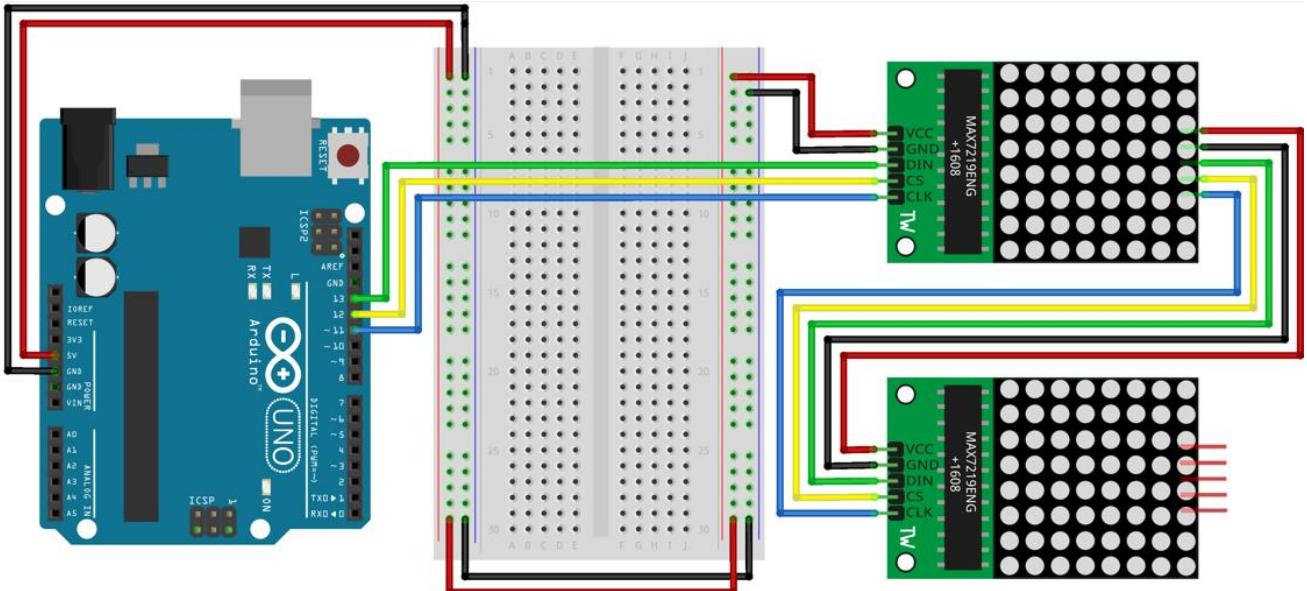
Note: Bxxxxxxxxx, betyder at tallet er skrevet som et binært tal, dvs. at 1 betyder LED'en er tændt og 0 betyder LED'en slukket.

## 6 LED matrix 8x8 chain

Flere LED 8x8 matrixmoduler kan sættes sammen i en kæde.



## 6.1 Ledningsdiagrammer



## 6.2 Arduino bibliotek

Arduino bibliotek – se "single" lige oven over.

## 6.3 Software eksempel

```
#include <MD_MAX72xx.h>

#define SCREEN_NUM 2

#define DATA_PIN 13
#define CS_PIN 12
#define CLK_PIN 11

MD_MAX72XX mx = MD_MAX72XX(MD_MAX72XX::PAROLA_HW, DATA_PIN, CLK_PIN, CS_PIN,
SCREEN_NUM);

void setup() {
    mx.begin();
}

void loop(){
    scissor(0);
    mx.clear(1);
    delay(1000);

    scissor(1);
    mx.clear(0);
    delay(1000);
}

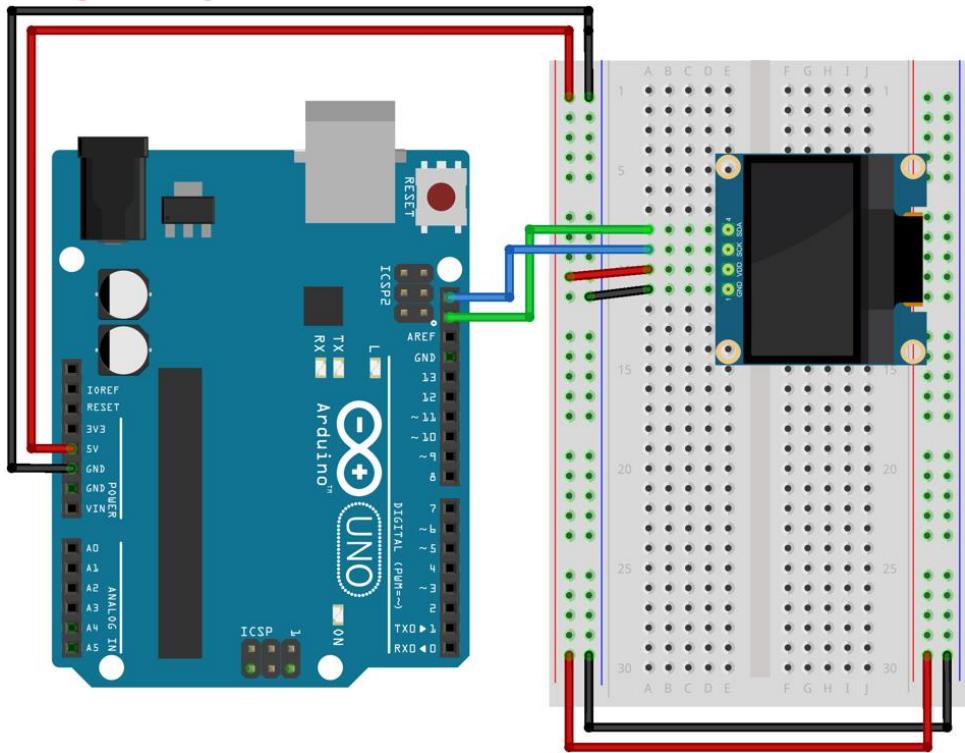
void scissor(int screen) {
    byte rows[8] = {
        B01000001,
        B10100010,
        B01100100,
        B00011000,
        B00011000,
        B01100100,
        B10100010,
        B01000001,
    };
    for(int i = 0; i <= 7; i++) {
        mx.setRow(screen, i, rows[i]);
    }
}
```

## 7 OLED 0.96" display

Lille 0.96" LCD skærm (købt som OLED), med 128x64 pixels i en farve.



## 7.1 Ledningsdiagrammer



## 7.2 Arduino bibliotek

Følgende bibliotek skal installeres: *Adafruit GFX*.

[Adafruit GFX Library](#) by Adafruit

1.10.4 installed

Install this library in addition to the display library for your hardware. Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from.

[More info](#)

Følgende bibliotek skal installeres: *Adafruit SSD1306*.

[Adafruit SSD1306](#) by Adafruit

...

2.4.2 installed

SSD1306 oled driver library for monochrome 128x64 and 128x32 displays SSD1306 oled driver library for monochrome 128x64 and 128x32 displays

[More info](#)

## 7.3 Software eksempel

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

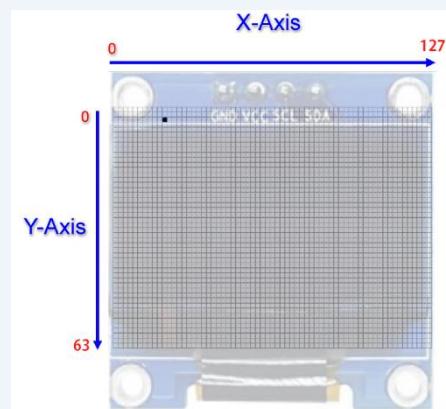
Adafruit_SSD1306 display(128, 64, &Wire);

void setup() {
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.setTextSize(2);
    display.setTextColor(true);
}

void loop() {
    display.clearDisplay();

    display.drawLine(10, 15, 117, 15, true);
    display.setCursor(10, 25);
    display.println("MAKERCAMP");
    display.drawLine(10, 47, 117, 47, true);

    display.display();
}
```

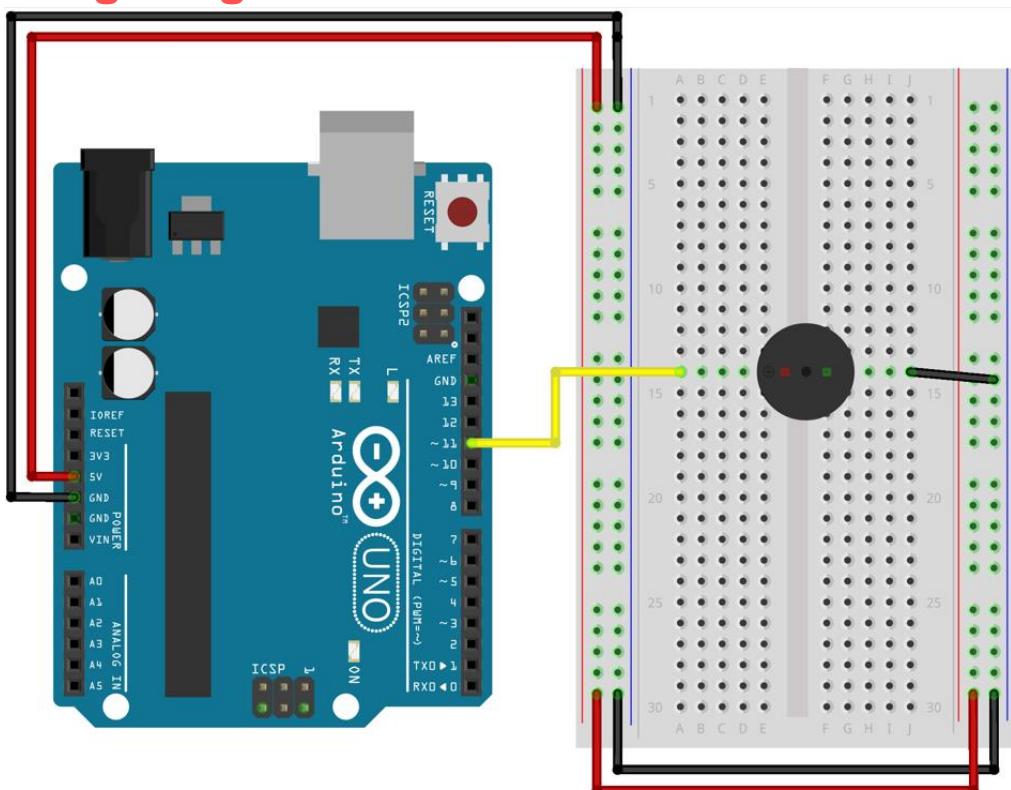


## 8 Buzzer

Buzzer'en findes i både en aktiv og passiv version. Vi har kun den passive version som er beskrevet her. Den passive version fungerer som en lille højttaler, som skal have en frekvens for at sige noget.



### 8.1 Ledningsdiagrammer



## 8.2 Software eksempel

```
#define BUZZER_PIN 11

void setup() {
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  tone(BUZZER_PIN, 440);
  delay(1000);

  noTone(BUZZER_PIN);
  delay(200);

  tone(BUZZER_PIN, 288);
  delay(1000);

  noTone(BUZZER_PIN);
  delay(200);
}
```

## 9 Relæ styring

Dette relæ kan tænde og slukke for andre komponenter.  
Der skal også bruges en modstand på  $220\ \Omega$ .

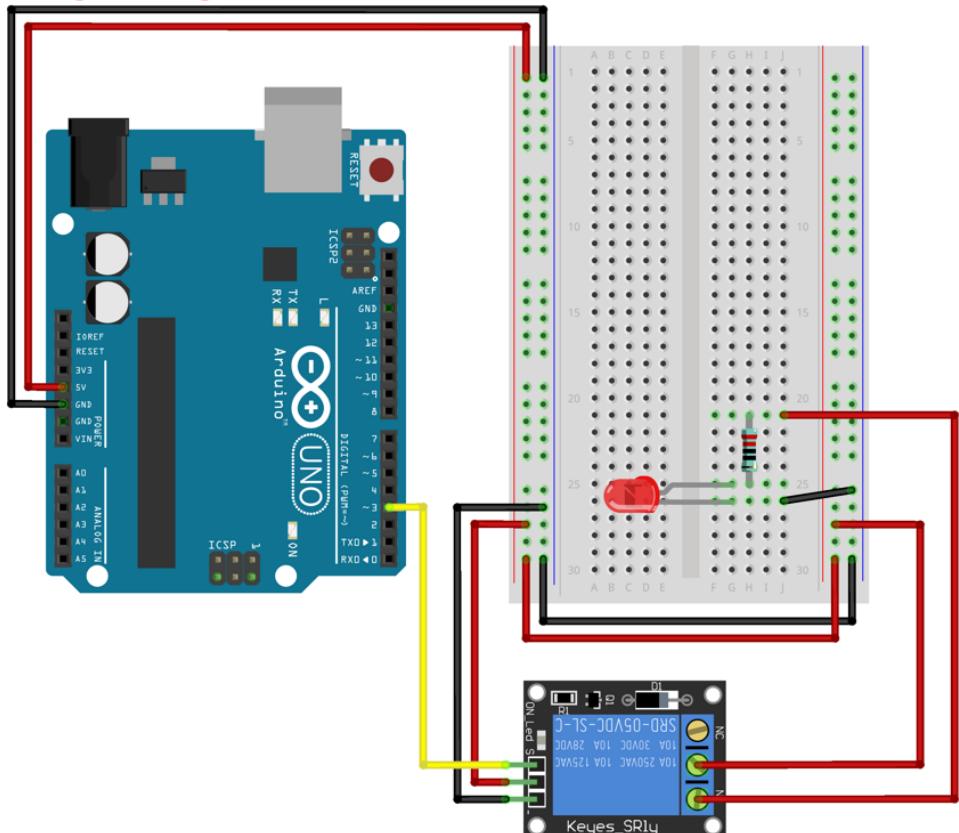


Modstand  $220\ \Omega$



To røde streger.

## 9.1 Ledningsdiagrammer



## 9.2 Software eksempel

```
#define RELAY_PIN 3

void setup() {
    pinMode(RELAY_PIN, OUTPUT);
}

void loop() {
    digitalWrite(RELAY_PIN, HIGH);
    delay(500);
    digitalWrite(RELAY_PIN, LOW);
    delay(500);
}
```

## 10 Knap, kontakt

Arduinoen kan aflæse en tilstanden af en knap eller kontaktakt. Vi har flere forskellige typer at vælge imellem. Der skal også bruges en modstand på 10 kΩ.



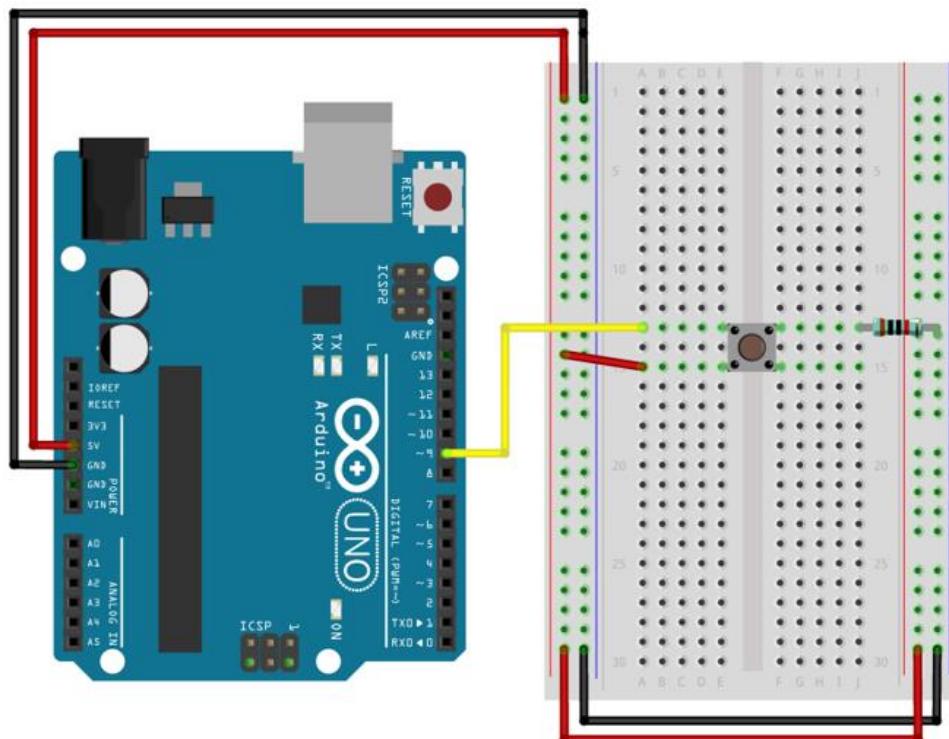


Modstand 10 kΩ



Én rød streg.

## 10.1 Ledningsdiagrammer



## 10.2 Software eksempel

```
#define BUTTON_PIN 9

int buttonState;

void setup() {
  pinMode(BUTTON_PIN, INPUT);
  Serial.begin(9600);
}

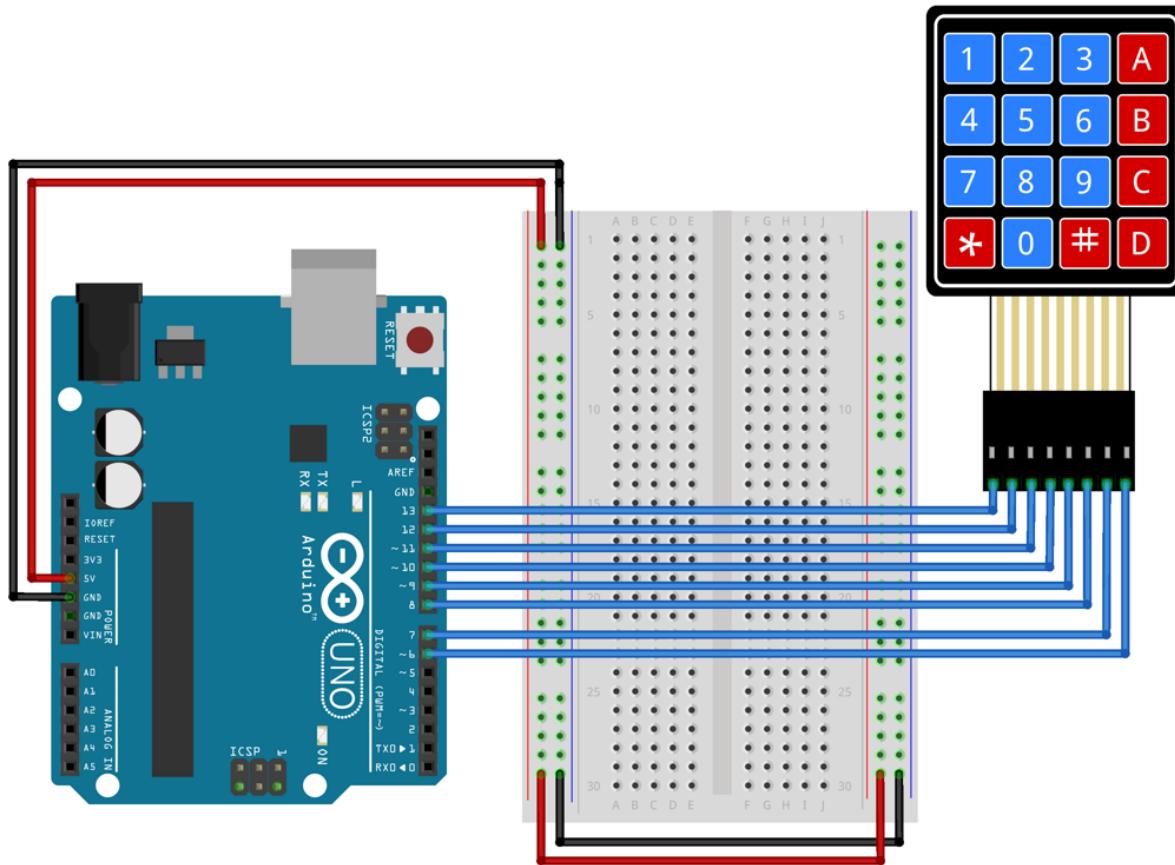
void loop() {
  buttonState = digitalRead(BUTTON_PIN);
  Serial.println(buttonState);
}
```

# 11 Keypad 4x4

Et fladt keypad med 16 taster.



## 11.1 Ledningsdiagrammer



## 11.2 Arduino bibliotek

Følgende bibliotek skal installeres: *keypad*.

**Keypad** by Mark Stanley, Alexander Brevig

[3.1.1 installed](#)

As of version 3.0 it now supports multiple keypresses. This library is based upon the Keypad Tutorial. It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user. Keypad is a library for using matrix style keypads with the Arduino.

[More info](#)

## 11.3 Software eksempel, aflæs enkelt cifre

```
#include <Keypad.h>

char keyMap[4][4] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPins[4] = {2, 3, 4, 5};
byte colPins[4] = {6, 7, 8, 9};

Keypad customKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, 4, 4);

void setup(){
    Serial.begin(9600);
}

void loop() {
    char key;
    long num = readNumber(&key);

    Serial.print(num);
    Serial.print(" ");
    Serial.println(key);
}

long readNumber(char *lastKey) {
    char keyPressed;
    long num = 0;

    for (;;) {
        keyPressed = customKeypad.getKey();
        if (keyPressed){
            if (keyPressed >= '0' && keyPressed <= '9') {
                num = num*10 + (keyPressed)-48;
            }
            else {
                *lastKey = keyPressed;
                return num;
            }
        }
    }
}
```

## 11.4 Software eksempel, aflæs kode som tal

```
#include <Keypad.h>

char keyMap[4][4] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPins[4] = {2, 3, 4, 5};
byte colPins[4] = {6, 7, 8, 9};

Keypad customKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, 4, 4);

void setup(){
    Serial.begin(9600);
}

void loop() {
    char key;
    long num = readNumber(&key);

    Serial.print(num);
    Serial.print(" ");
    Serial.println(key);
}

long readNumber(char *lastKey) {
    char keyPressed;
    long num = 0;

    for (;;) {
        keyPressed = customKeypad.getKey();
        if (keyPressed){
            if (keyPressed >= '0' && keyPressed <= '9') {
                num = num*10 + (keyPressed)-48;
            }
            else {
                *lastKey = keyPressed;
                return num;
            }
        }
    }
}
```

## 11.5 Software eksempel, aflæs kode som String

```
#include <Keypad.h>

char keyMap[4][4] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPins[4] = {2, 3, 4, 5};
byte colPins[4] = {6, 7, 8, 9};

Keypad customKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, 4, 4);

void setup() {
    Serial.begin(9600);
}

void loop() {
    String myString = readWord();
    Serial.println(myString);
}

String readWord() {
    char keyPressed;
    String password = "";

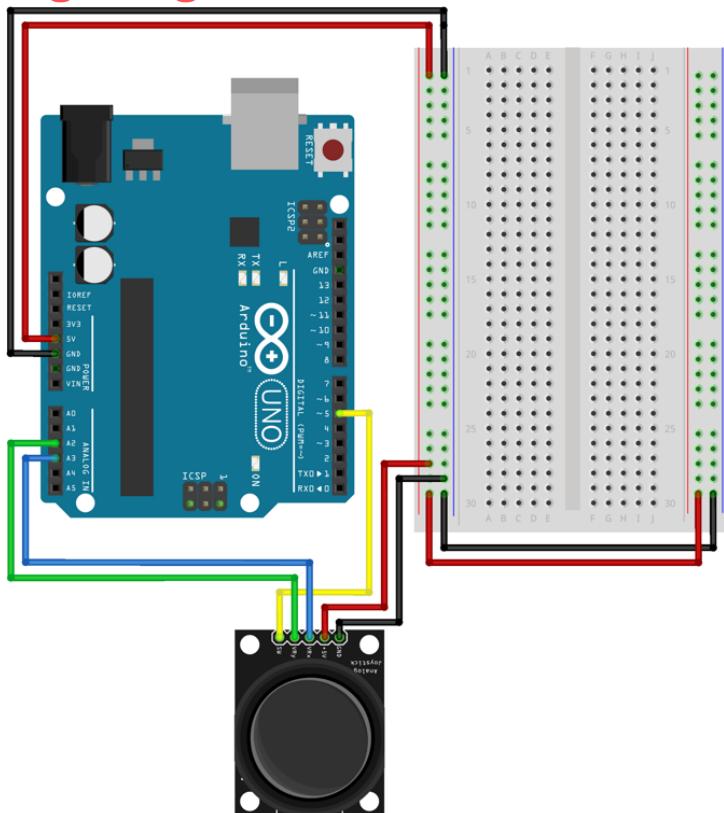
    for (;;) {
        keyPressed = customKeypad.getKey();
        if (keyPressed){
            if (keyPressed != '#') { //Vælg en tegn som stopper opsamling.
                password.concat(keyPressed);
            }
            else {
                return password;
            }
        }
    }
}
```

## 12 Joystick

Analog joystick med trykknap.



## 12.1 Ledningsdiagrammer



## 12.2 Software eksempler

```
#define JOY_X_PIN A2
#define JOY_Y_PIN A3
#define JOY_BUTTON_PIN 5

int xValue;
int yValue;
int joyButtonState;

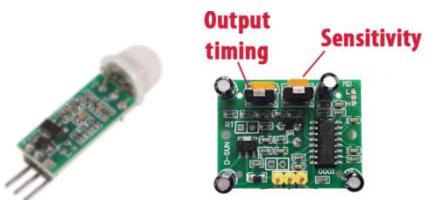
void setup() {
    Serial.begin(9600);
    pinMode(JOY_X_PIN, INPUT);
    pinMode(JOY_Y_PIN, INPUT);
    pinMode(JOY_BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
    xValue = analogRead(JOY_X_PIN);
    yValue = analogRead(JOY_Y_PIN);
    joyButtonState = !digitalRead(JOY_BUTTON_PIN);

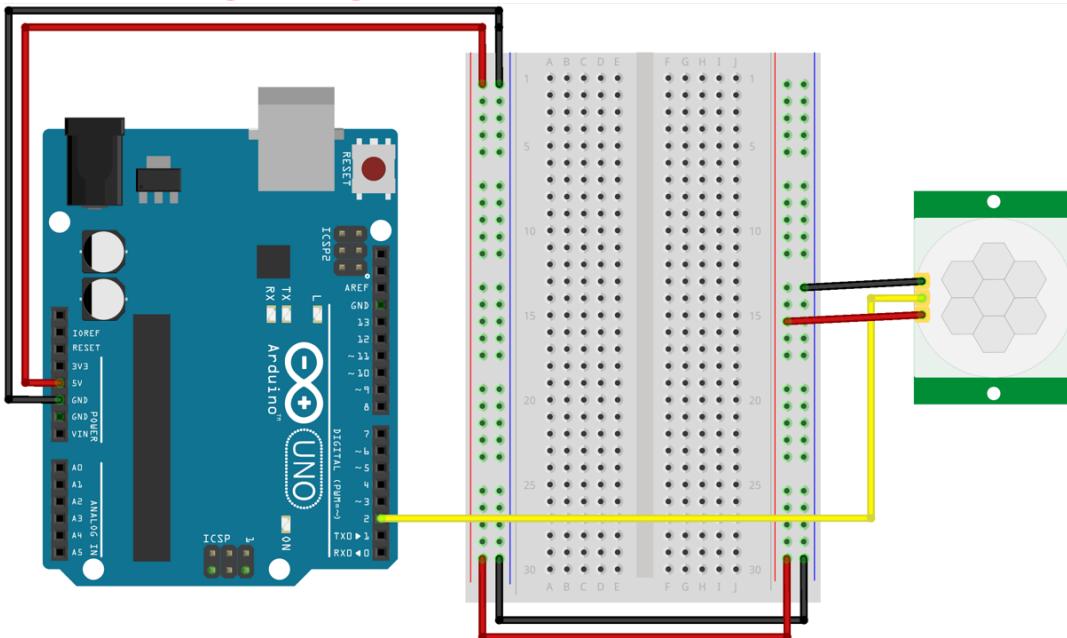
    Serial.print("X: ");
    Serial.print(xValue);
    Serial.print(" ");
    Serial.print("Y: ");
    Serial.print(yValue);
    Serial.print(" ");
    Serial.print("Button: ");
    Serial.print(joyButtonState);
    Serial.println();
}
```

# 13 PIR motion sensor

En PIR (Passive InfraRed sensor) kan måle bevægelse. Vi har flere forskellige typer der alle forbindes på samme måde. På nogle af dem kan der indstilles følsomheden og hvor hurtigt de skal reagere på bevægelse.



## 13.1 Ledningsdiagrammer



## 13.2 Software eksempel

```
#define MOTION_PIN 2

bool motionState;

void setup() {
  Serial.begin(9600);
  pinMode(MOTION_PIN, INPUT);
}

void loop() {
  motionState = digitalRead(MOTION_PIN);

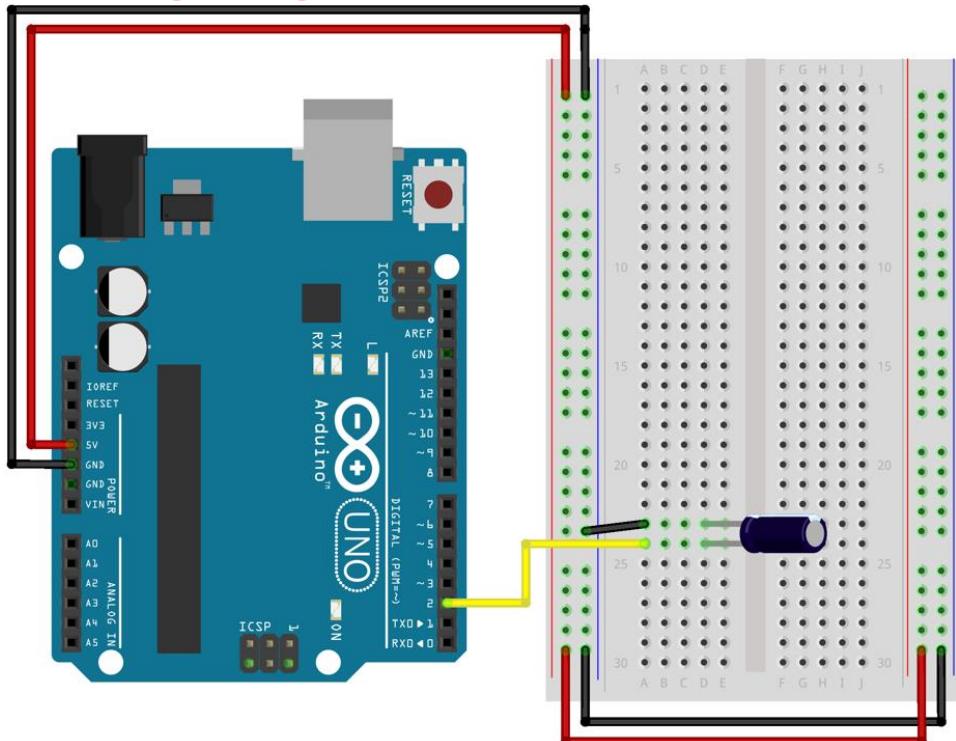
  if(motionState == true) {
    Serial.println("bevægelse!");
  } else {
    Serial.println("INGEN bevægelse");
  }
}
```

# 14 Tilt/ryste sensor

Denne sensor kan mekanisk mærke hvis den tilter. Den ligger fysisk en kugle inden i den som bevæger sig og fungerer som en kontakt. Det kan derfor være en fordel at montere sensoren på nogle ledninger.



## 14.1 Ledningsdiagrammer



## 14.2 Software eksempel

```
#define TILT_PIN 2

int tiltState;

void setup() {
    Serial.begin(9600);
    pinMode(TILT_PIN, INPUT_PULLUP);
}

void loop() {
    tiltState = digitalRead(TILT_PIN);
    Serial.println(tiltState);
}
```

## 15 Lysfølsom modstand

En lysfølsom modstand (fotomodstand) ændrer sin modstand alt efter hvor meget lys den opfanger. Man kan derfor måle lysstyrke.

Man skal også bruge en modstand på  $10\text{ k}\Omega$ .

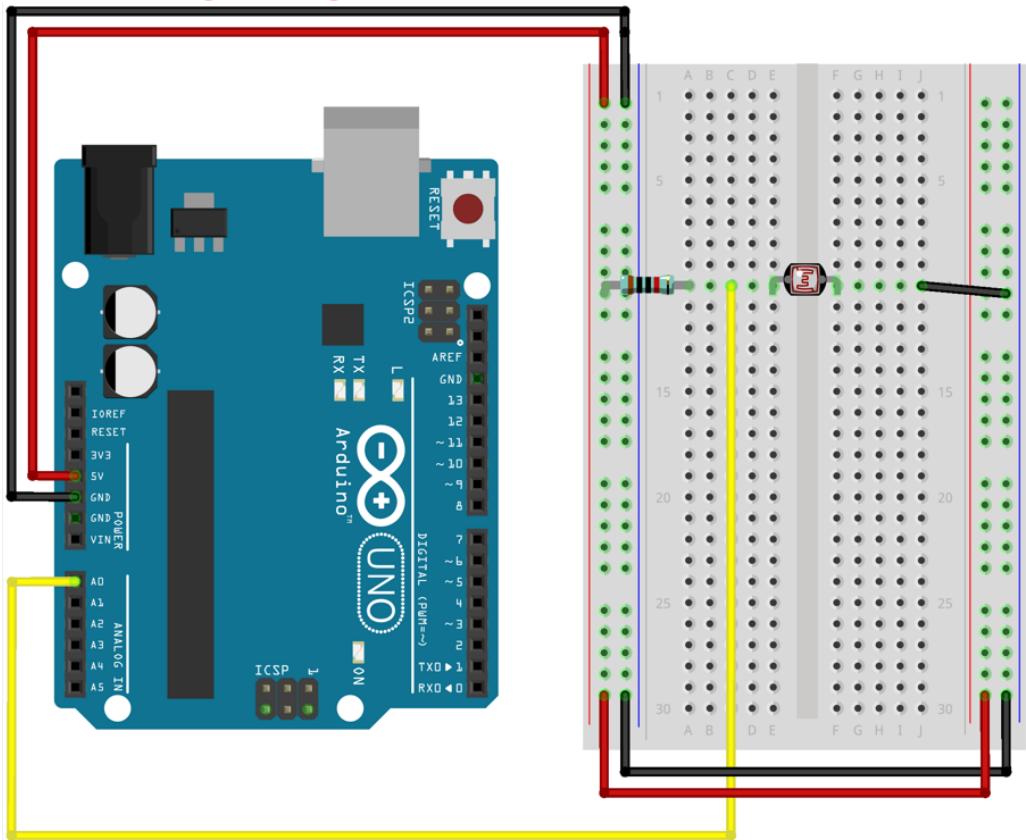


Modstand  $10\text{ k}\Omega$



Én rød streg.

## 15.1 Ledningsdiagrammer



## 15.2 Software eksempel

```
#define LIGHT_PIN A0

int lightValue;

void setup() {
    Serial.begin(9600);
    pinMode(LIGHT_PIN, INPUT);
}

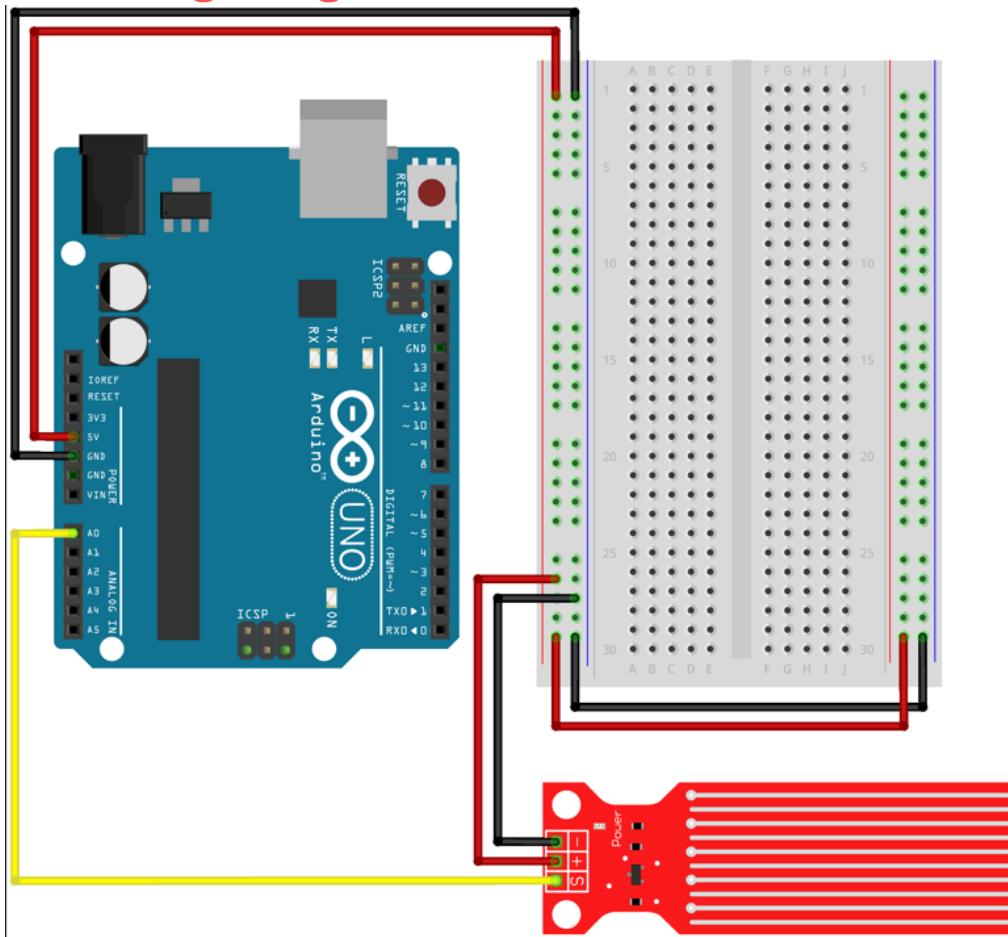
void loop() {
    lightValue = analogRead(LIGHT_PIN);
    Serial.println(lightValue);
}
```

## 16 Waterlevel sensor

Sensor som kan måle vandhøjde.



## 16.1 Ledningsdiagrammer



## 16.2 Software eksempel

```
#define WATER_PIN A0

int waterValue;

void setup() {
    pinMode(WATER_PIN, INPUT);
    Serial.begin(9600);
}

void loop() {
    waterValue = analogRead(WATER_PIN);

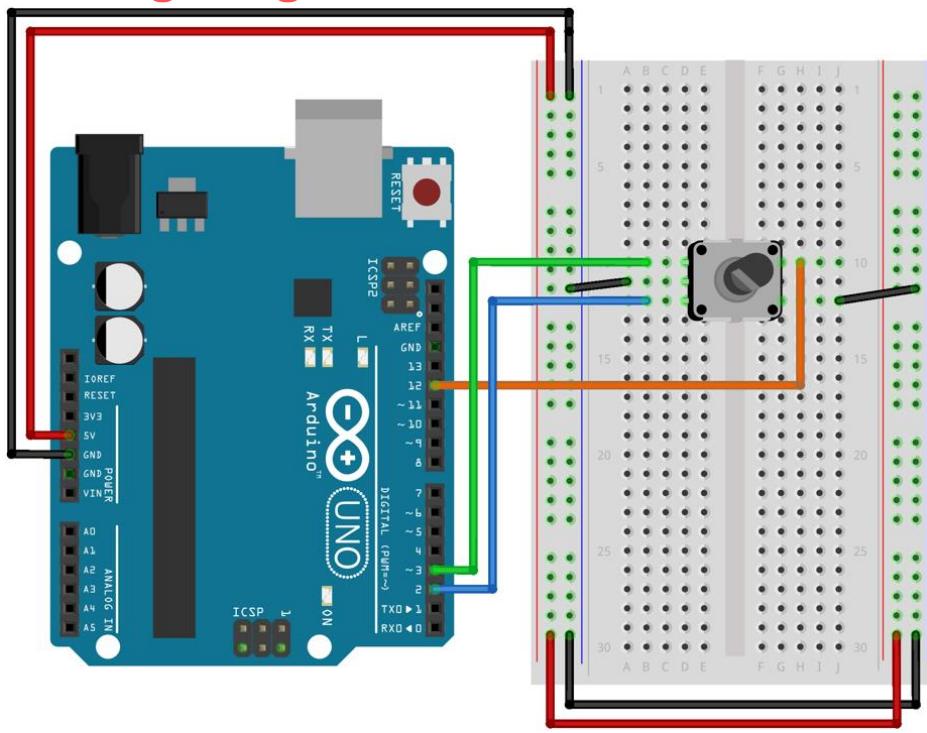
    if(waterValue > 650) {
        Serial.print("Full   ");
    } else if(waterValue > 550) {
        Serial.print("Half   ");
    } else {
        Serial.print("Empty   ");
    }
    Serial.println(waterValue);
}
```

# 17 Rotary encoder

En rotary encoder er en drejeknap som kan registrere hvor mange 'klik' den drejes og i hvilken retning den drejes. Den har også en knap funktion, når man trykker på den.



## 17.1 Ledningsdiagrammer



## 17.2 Arduino bibliotek

Følgende bibliotek skal installeres: *Rotary*.

[Rotary by KAthiR](#)

[1.0.0 installed](#)

It allows you to use callback functions to track rotation. This will reduce and simplify your source code significantly. Tested with Arduino and ESP8266. ESP8266/Arduino Library for reading rotary encoder values.

[More info](#)

## 17.3 Software eksempel

```
#include "Rotary.h";

#define ROTARY_PIN1 2
#define ROTARY_PIN2 3
#define BUTTON_PIN 12

#define MIN_POS      0
#define MAX_POS     10
#define START_POS    5

Rotary myEncoder;

int buttonState;
int encoderPosition = START_POS;

void setup() {
  Serial.begin(9600);
  pinMode(ROTARY_PIN1, INPUT_PULLUP);
  pinMode(ROTARY_PIN2, INPUT_PULLUP);
  pinMode(BUTTON_PIN, INPUT_PULLUP);

  myEncoder.begin(ROTARY_PIN1, ROTARY_PIN2, 4, MIN_POS, MAX_POS, START_POS, 1);
  myEncoder.setChangedHandler(rotate);
}

void loop() {
  myEncoder.loop();
  buttonState = !digitalRead(BUTTON_PIN);

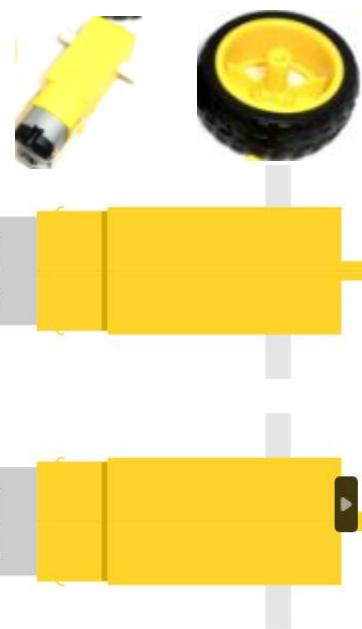
  if(buttonState == HIGH) {
    Serial.println("Click");
  }
}

void rotate(Rotary& myEncoder) {
  encoderPosition = myEncoder.getPosition();

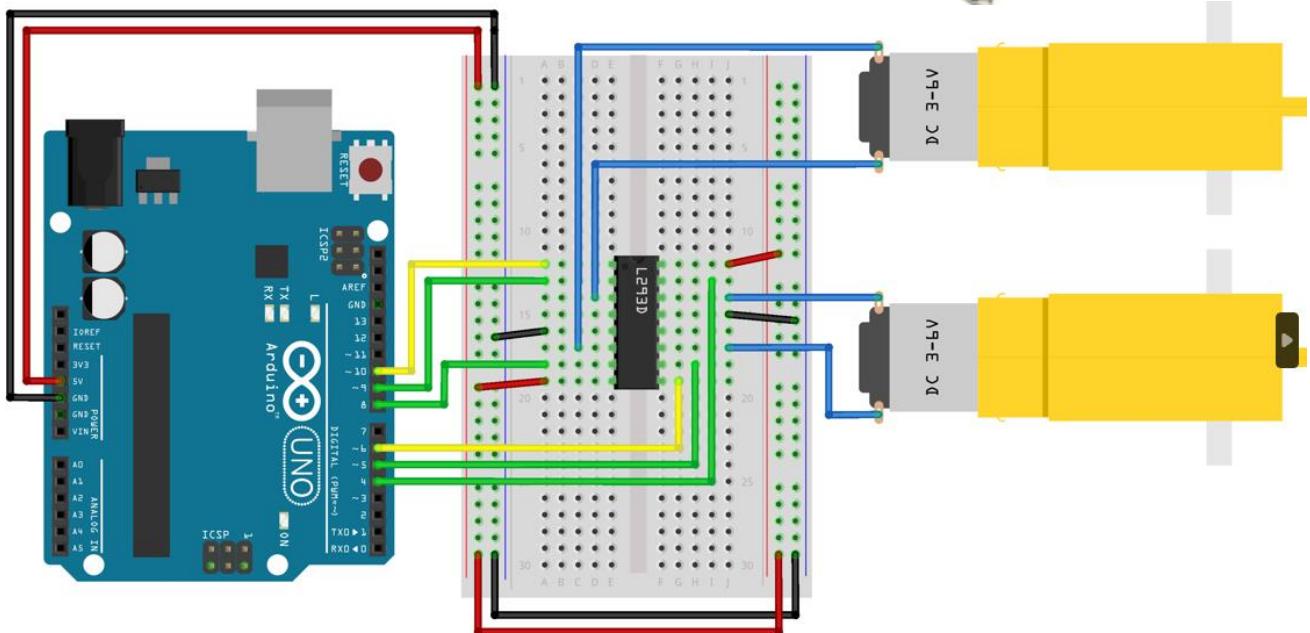
  Serial.print("position: ");
  Serial.print(encoderPosition);
  Serial.println();
}
```

# 18 Motor med L293D

Denne motor kan bruges til at lave en bil med. I eksemplet bruges 2 motorer.



## 18.1 Ledningsdiagrammer



## 18.2 Software eksempel

```
#define ENA 6
#define ENB 10
#define IN1 5
#define IN2 4
#define IN3 8
#define IN4 9

void setup() {
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
}

void loop() {
    forward();
    delay(2000);

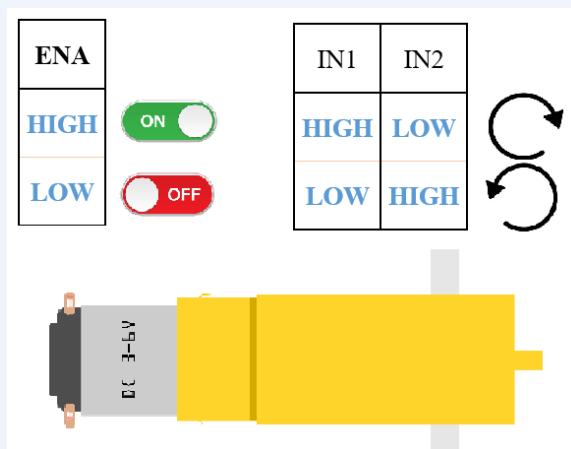
    backward();
    delay(2000);

    stopMotor();
    delay(2000);
}

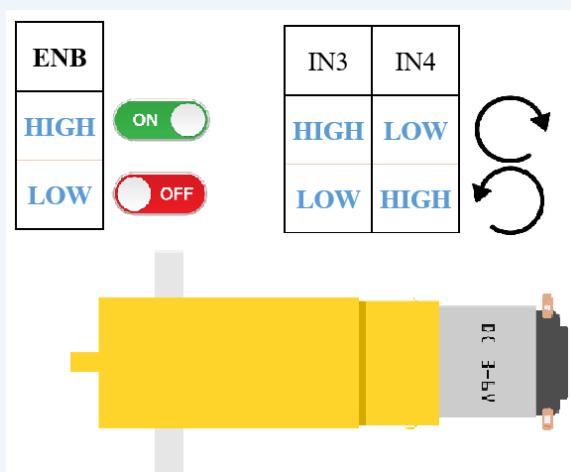
void forward() {
    digitalWrite(ENA, HIGH);
    digitalWrite(ENB, HIGH);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void backward() {
    digitalWrite(ENA, HIGH);
    digitalWrite(ENB, HIGH);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void stopMotor() {
    digitalWrite(ENA, LOW);
    digitalWrite(ENB, LOW);
}
```



**Motor 1**



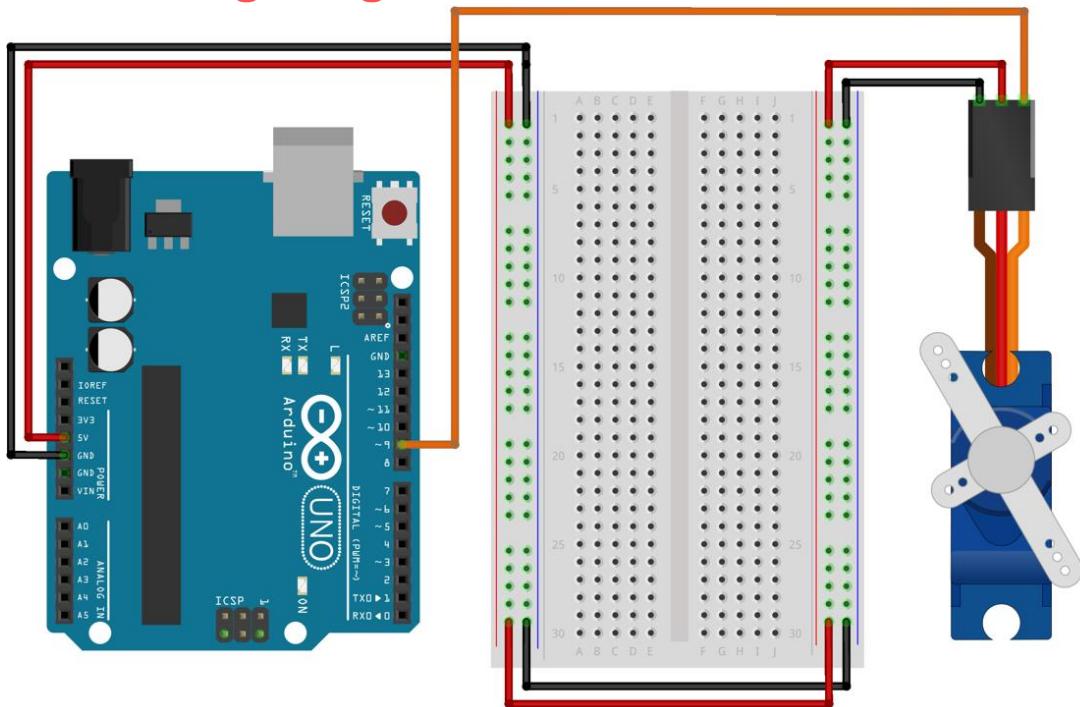
**Motor 2**

## 19 Servo

En servo er en motor, hvor man kan styre position med en vinkel mellem 0 og 180 grader.



## 19.1 Ledningsdiagrammer



## 19.2 Software eksempel

```
#include <Servo.h>

#define SERVO_PIN 9

Servo myservo;

int position1 = 0;
int position2 = 180;

void setup() {
  myservo.attach(SERVO_PIN);
}

void loop() {
  myservo.write(position1);
  delay(1000);

  myservo.write(position2);
  delay(1000);
}
```

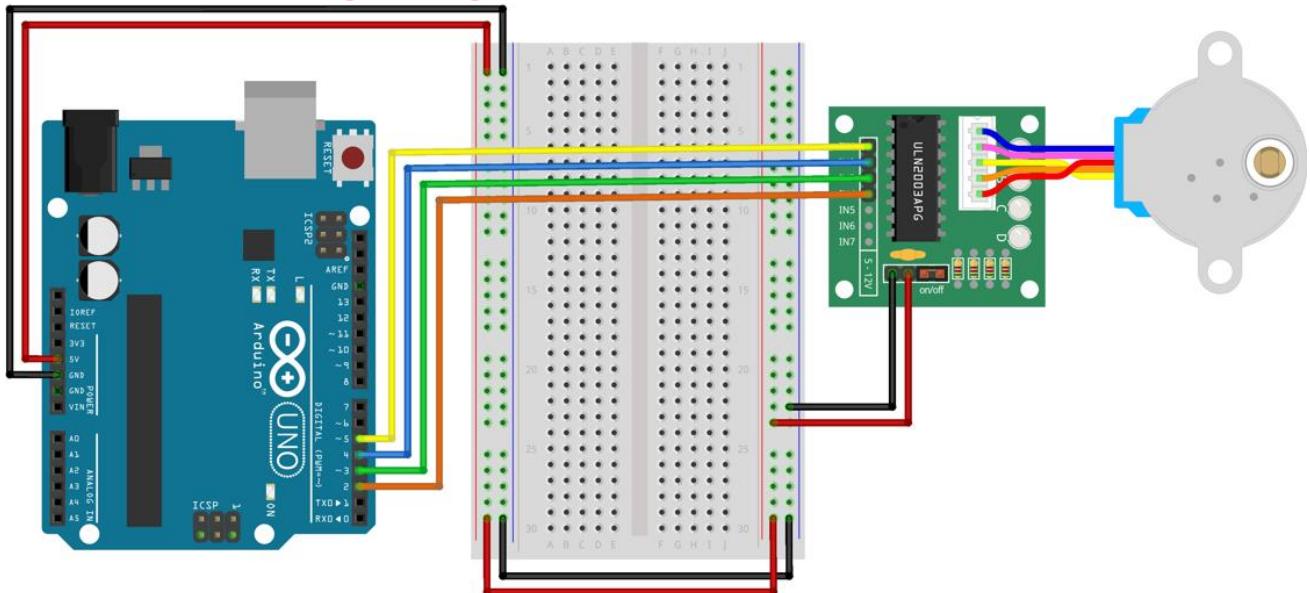
## 20 Stepmotor

En stepermotor er en motor, hvor man kan styre hvor mange steps den skal dreje i en given retning, dvs. at man fx kan få den til at dreje præcist 73 steps til højre.

Motoren bruger 2048 steps på én omdrejning.



## 20.1 Ledningsdiagrammer



## 20.2 Arduino bibliotek

Følgende bibliotek skal installeres: *TinyStepper\_28BYJ\_48*.

[TinyStepper\\_28BYJ\\_48](#) by S. Reifel

1.0.0 installed

Easy to use functions for controlling one or more 28BYJ-48 stepper motors. Uses speed and acceleration control for smooth operations and faster rotations. Requires a ULN2003 Driver Board. Tiny stepper motor 28BYJ-48 control library for Arduino.

[More info](#)

## 20.3 Software eksempel

```
#include <TinyStepper_28BYJ_48.h>

#define IN1 2
#define IN2 3
#define IN3 4
#define IN4 5

TinyStepper_28BYJ_48 stepper;

void setup() {
    Serial.begin(9600);
    stepper.connectToPins(IN1, IN2, IN3, IN4);
    stepper.setSpeedInStepsPerSecond(256);
    stepper.setAccelerationInStepsPerSecondPerSecond(512);
}

void loop() {
    stepper.moveRelativeInSteps(512);
    disableMotor();
    delay(1000);

    stepper.moveRelativeInSteps(-512);
    disableMotor();
    delay(1000);
}

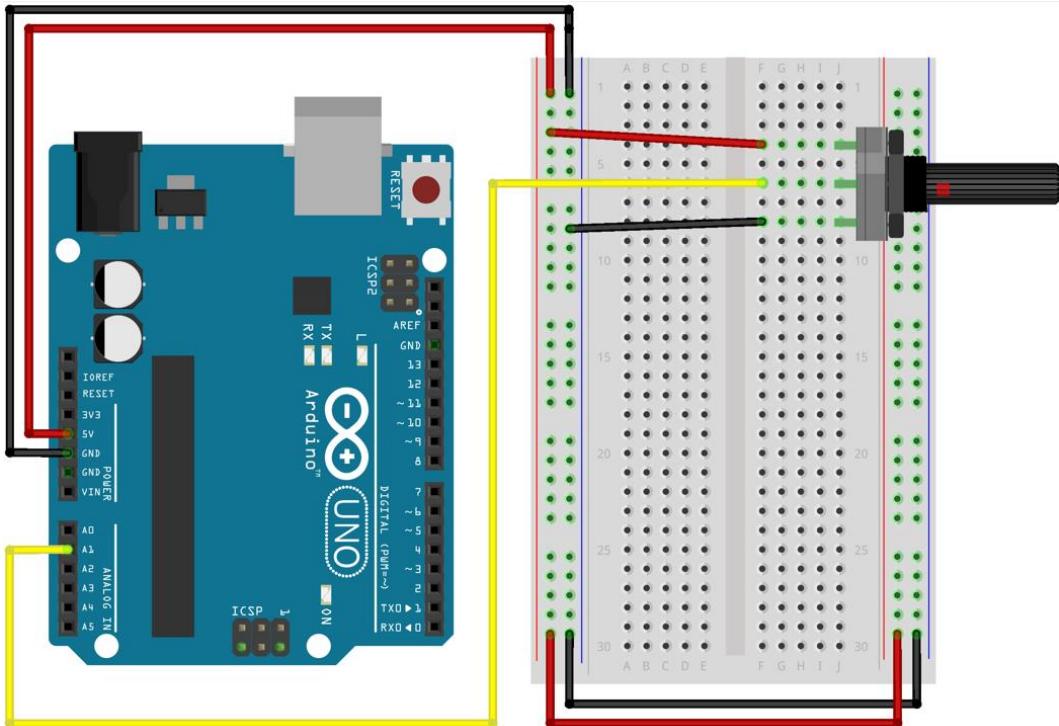
void disableMotor() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}
```

## 21 Potentiometeret

Potentiometeret er en variabel modstand, som kan indstilles ved at dreje på den. Den kaldes også et potmeter.



## 21.1 Ledningsdiagrammer



## 21.2 Software eksempel

```
#define POT_PIN A1

int potValue;

void setup() {
    Serial.begin(9600);
    pinMode(POT_PIN, INPUT);
}

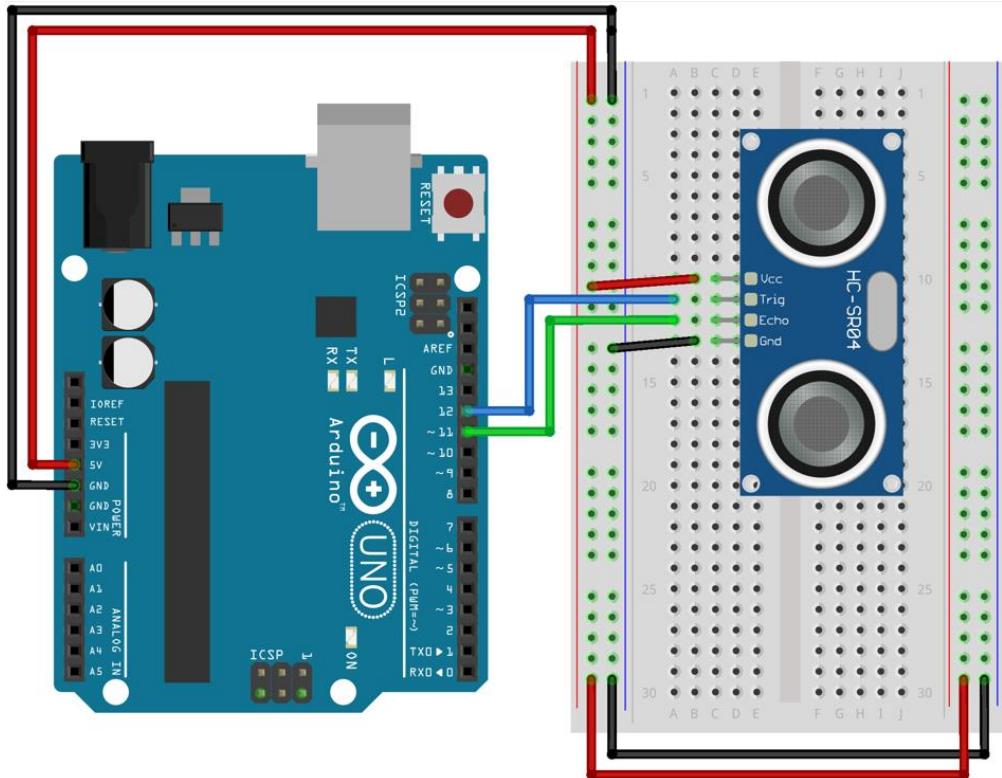
void loop() {
    potValue = analogRead(POT_PIN);
    Serial.println(potValue);
}
```

## 22 Afstandssensor

Ultralydssensoren kan måle en afstand fra få centimeter til flere meter, med en nøjagtighed på ca. 1 cm.



## 22.1 Ledningsdiagrammer



## 22.2 Software eksempel

```
#define TRIG_PIN 12
#define ECHO_PIN 11

int distance;

void setup() {
    Serial.begin(9600);
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
}

void loop() {
    distance = readUltrasonic();

    Serial.print("afstand: ");
    Serial.println(distance);
}

int readUltrasonic() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);

    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);

    digitalWrite(TRIG_PIN, LOW);
    long ultrasonicMilliSec = pulseIn(ECHO_PIN, HIGH);

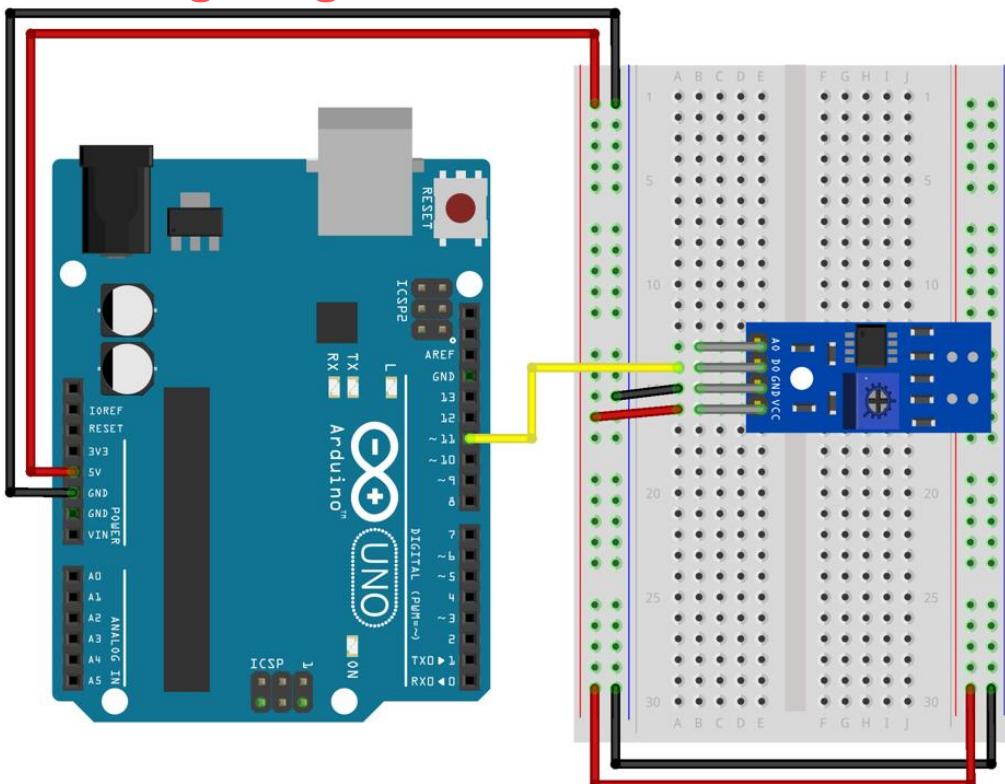
    return ultrasonicMilliSec * 0.034 / 2;
}
```

# 23 Linjesensor (infrarød)

Sensoren kan "se" en linje og det er dermed muligt at lave en bil som følger en linje på gulvet. Man kan indstille følsomheden på modulet. Det vil være en fordel at bruge to sensorer, så man kan se, hvilken side af linjen som bilen er på vej væk fra.



## 23.1 Ledningsdiagrammer



## 23.2 Software eksempel

Simpelt eksempel på hvordan man aflæser sensoren.

```
#define IR_PIN 11

bool irState;

void setup() {
  Serial.begin(9600);
  pinMode(IR_PIN, INPUT);
}

void loop() {
  irState = digitalRead(IR_PIN);
  Serial.println(irState);
}
```

Nedenfor ser du et eksempel hvordan man laver en bil, som følger en sort linje. Man skal bruge 2 IR sensorer, som skal sidde ved siden af hinanden foran på bilen og pege nedad, så de kan se den sorte linje. Man skal selv skrive den kode som mangler. Hint, se koden for "Motor med L293D".

```

#define IR1_PIN 11
#define IR2_PIN 12

void setup() {
    pinMode(IR1_PIN, INPUT);
    pinMode(IR2_PIN, INPUT);
}

void loop() {
    if (digitalRead(IR1_PIN)) {
        right();
    }
    else if (digitalRead(IR2_PIN)) {
        left();
    }
    else {
        forward();
    }
}

void right() {

}

void left() {

}

void forward() {
}

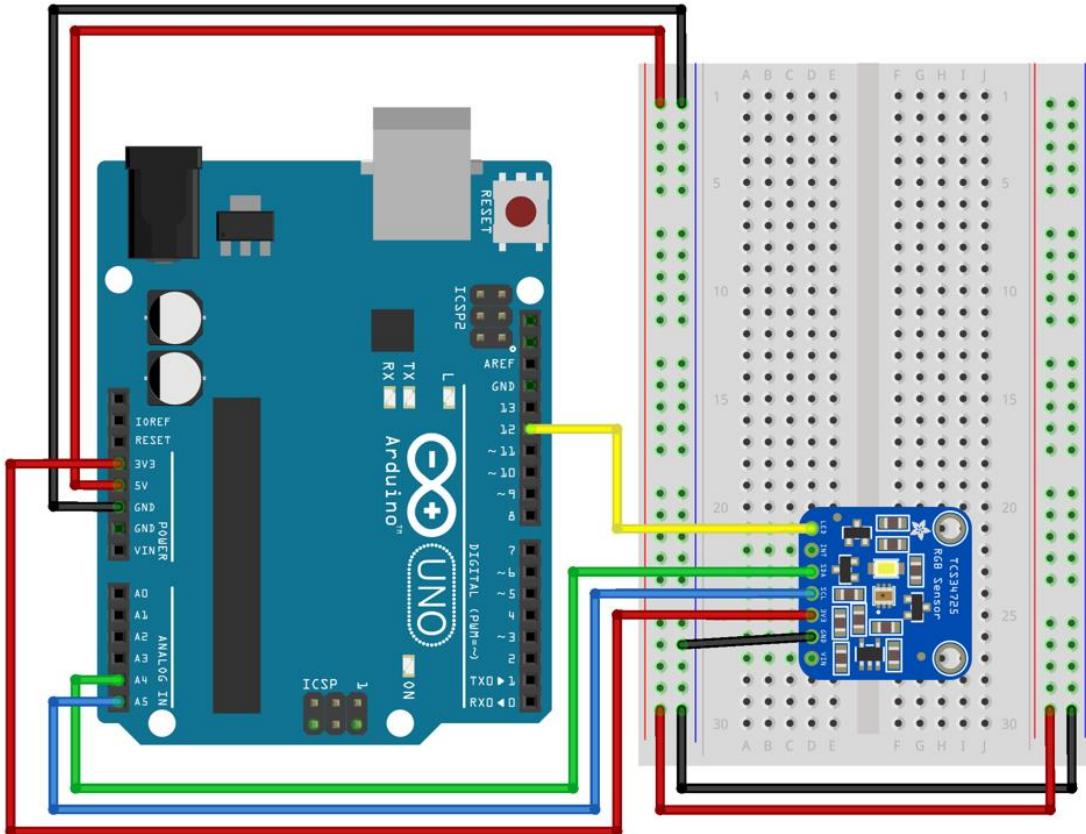
```

## 24 Farvesensor

Farvesensoren kan se hvilken farve den registrerer. Den har en LED som kan lyse op, så den lettere kan se farven.



## 24.1 Ledningsdiagrammer



## 24.2 Arduino bibliotek

Følgende bibliotek skal installeres: *Adafruit TCS34725*

### Adafruit TCS34725

by Adafruit Version 1.4.2 INSTALLED

Driver for Adafruit's TCS34725 RGB Color Sensor Breakout Driver for Adafruit's TCS34725 RGB Color Sensor Breakout

[More info](#)

## 24.3 Software eksempel

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"

Adafruit_TCS34725 tcs = Adafruit_TCS34725();

#define TCS_LED_PIN 12

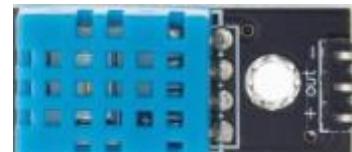
void setup(void) {
    Serial.begin(9600);
    if (!tcs.begin()) {
        Serial.println("No TCS34725 found");
        while (1);
    }
    pinMode(TCS_LED_PIN, OUTPUT);
    digitalWrite(TCS_LED_PIN, LOW); // Tænd LED
}

void loop(void) {
    float red, green, blue;

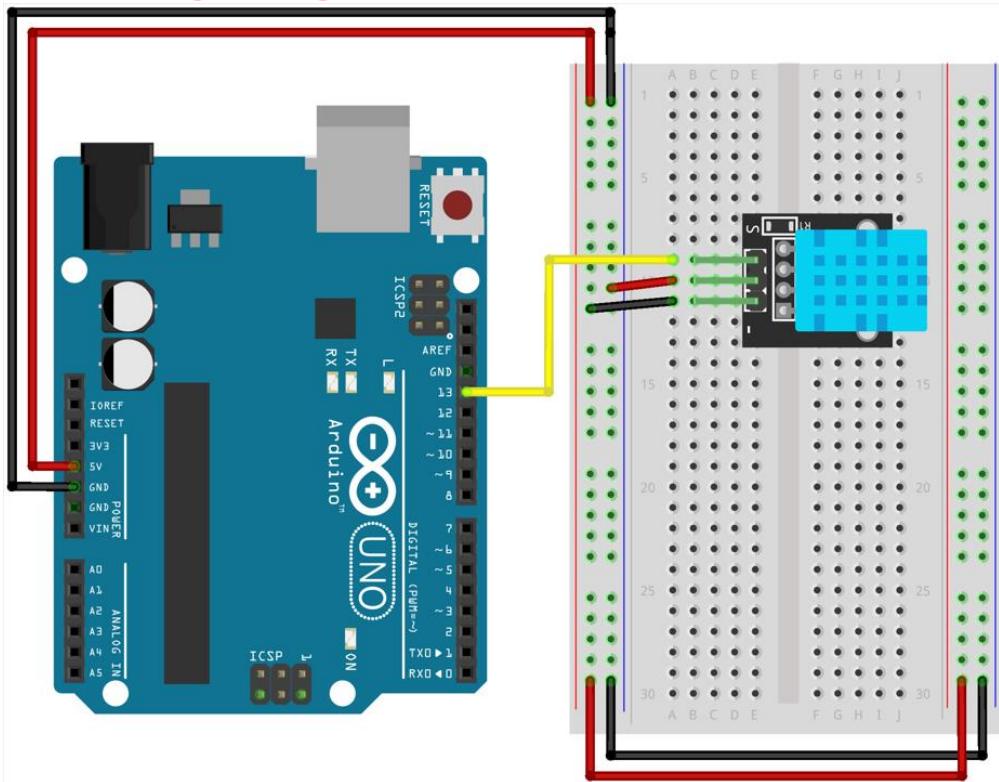
    tcs.getRGB(&red, &green, &blue);
    Serial.print("R: "); Serial.print(int(red)); Serial.print(" ");
    Serial.print("G: "); Serial.print(int(green)); Serial.print(" ");
    Serial.print("B: "); Serial.print(int(blue)); Serial.print(" ");
    Serial.println(" ");
}
```

## 25 Temperatursensor DHT11

Dette er en digital temperatursensor, som også kan måle fugtighed.  
Eksemplet viser dog kun hvordan man kan aflæse temperaturen.



## 25.1 Ledningsdiagrammer



## 25.2 Arduino bibliotek

Følgende bibliotek skal installeres: *DHTlib*

[DHTlib by Rob Tillaart <rob.tillaart@gmail.com>](#)

0.1.35 installed

AVR Optimized Library for DHT Temperature & Humidity Sensor on AVR only.

[More info](#)

## 25.3 Software eksempel

```
#include <dht.h>
#define DHT11_PIN      13
dht DHT;

int temp;

void setup() {
  Serial.begin(9600);
}

void loop() {
  DHT.read11(DHT11_PIN);
  temp = DHT.temperature;

  Serial.print(temp);
  Serial.println(" C");
  delay(1000);
}
```

# 26 Lydsensor

Lydsensoren kan måle om lydniveauet er over et vist niveau og dette niveau skal indstilles inden brug.

Den indstilles på følgende måde:

## Step 1:

Hvis LED2 (se billede) er **tændt**, så drej potentiometeret **MOD URET** indtil at du finder præcis det punkt hvor LED'en ligger på kanten mellem at tænde og slukke. (blinker)



Hvis LED2 er **slukket**, så drej potentiometeret **MED URET** indtil du finder præcis det punkt hvor LED'en ligger på kanten mellem at tænde og slukke. (blinker)

## Step 2:

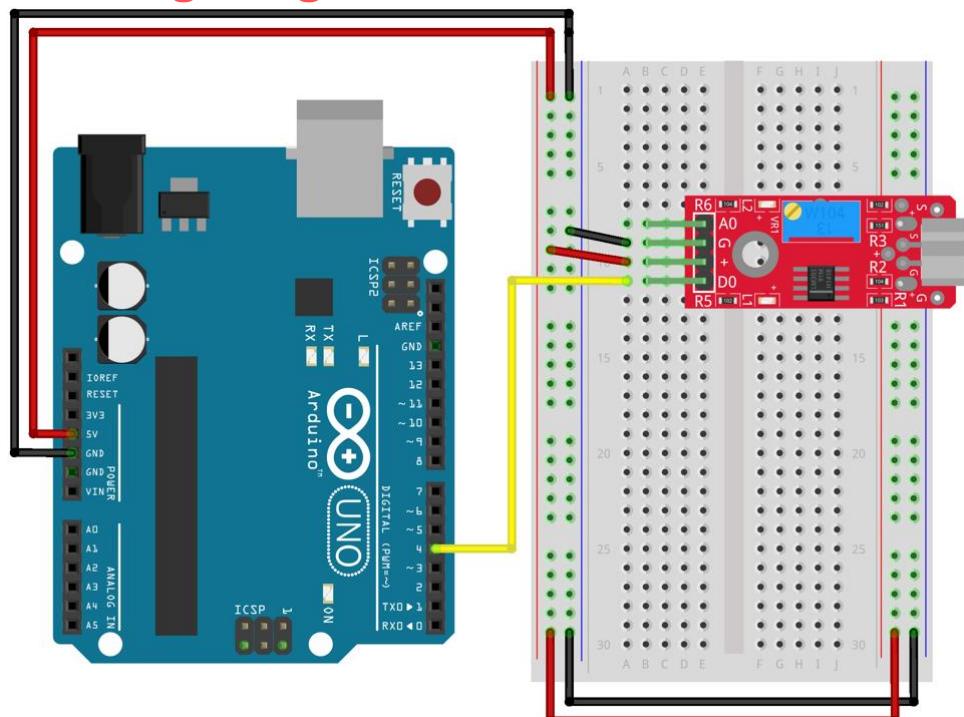
Drej potentiometeret **MOD URET** med meget små bevægelser, indtil LED'en konstant er slukket. Så snart LED'en slukker, STOP med at dreje!

## Test:

Hvis LED'en tænder og slukker når du klapper med hænderne foran lydsensoren eller blæser på den, er den indstillet korrekt.

Hvis LED'en ikke tænder i testen skal potentiometeret drejes en lille smule **MED URET**.

## 26.1 Ledningsdiagrammer



## 26.2 Software eksempel

```
#define SOUND_PIN 4

bool soundState;

void setup() {
  pinMode(SOUND_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  soundState = digitalRead(SOUND_PIN);
  if(soundState == true) {
    Serial.println("Lyd!");
  }
}
```

## 27 MP3 afspiller

MP3 afspilleren kan afspille MP3 filer fra et SD kort.



Der skal bruge to forskellige modstande, som er vigtige for at beskytte modulet fra at brænde af. De to modstande fås af instruktørerne og kan ses på nedenstående billede.



1 kΩ



2 kΩ

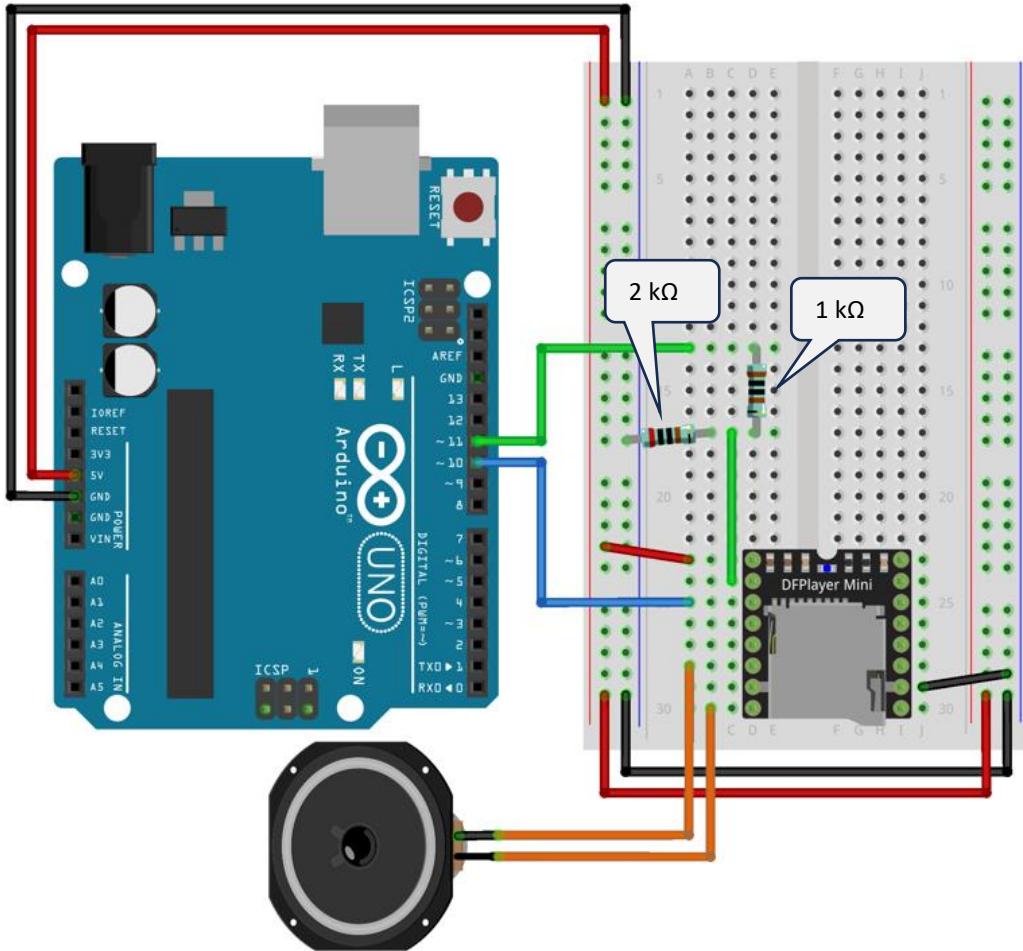


Ingen røde streger.



Én Rød streg Forrest.

## 27.1 Ledningsdiagrammer



## 27.2 Arduino bibliotek

Følgende bibliotek skal installeres: *DFRobotDFPPlayerMini*

**DFRobotDFPPlayerMini** by DFRobot

1.0.6 installed

Easy-to-use and reliable library for DFPlayer Mini Driver for DFPlayer Mini from DFRobot  
[More info](#)

## 27.3 Software eksempel

Navngiv altid MP3filer med tal. Navnet afgør hvilken rækkefølge filerne bliver afspillet i. Se billede nedenfor.

Filer som bruges i eksemplet, kan downloades her:

[https://drive.google.com/drive/folders/1Fk7EcSs3k0LDAawjWcV7z5mJYnPnXdQtb?usp=drive\\_link](https://drive.google.com/drive/folders/1Fk7EcSs3k0LDAawjWcV7z5mJYnPnXdQtb?usp=drive_link)

Andet gratis MP3 musik kan blandt andet findes her:

<https://pixabay.com/music/search/theme/background%20music/>

```

#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

#define TX_PIN 11
#define RX_PIN 10
SoftwareSerial softwareSerial(RX_PIN, TX_PIN);

DFRobotDFPlayerMini player;

void setup() {
  Serial.begin(9600);
  softwareSerial.begin(9600);
  player.begin(softwareSerial);
  player.volume(30); // 0 - 30
}

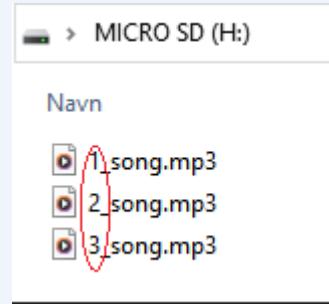
void loop() {
  player.play(1);
  delay(8000);

  player.play(2);
  delay(8000);

  player.play(3);
  delay(8000);

  player.pause();
  delay(5000);
}

```

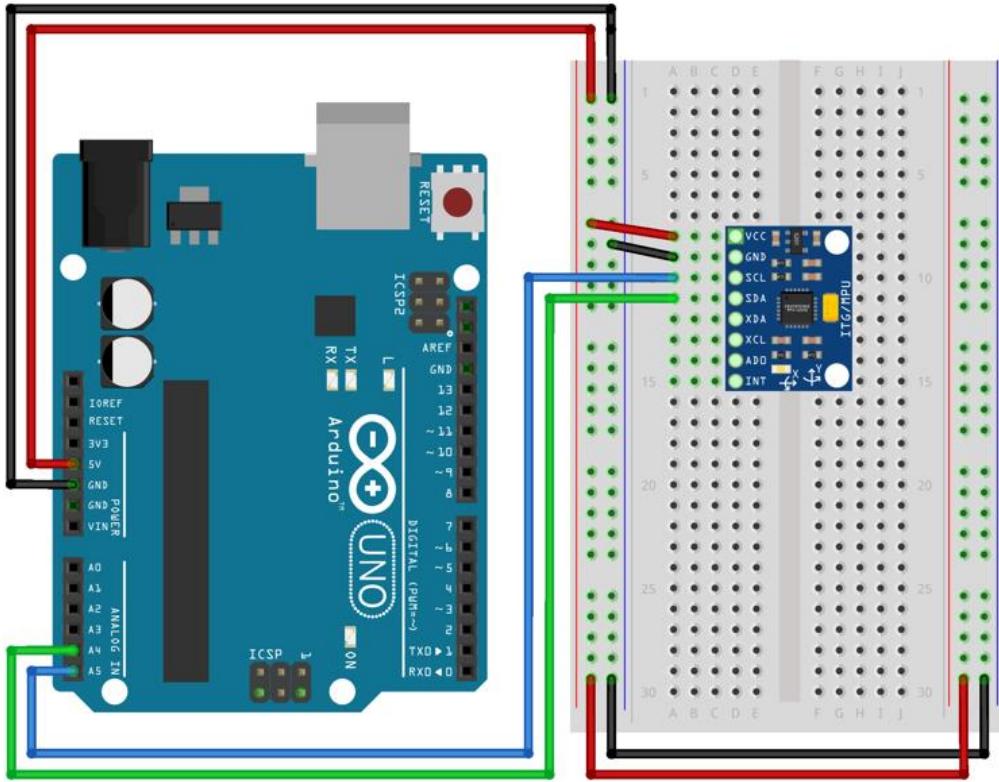


## 28 Gyro

Dette modul har én 3-akset gyro og et 3-akset accelerometer, dvs. at den kan måle bevægelser og vinkler i forhold til jorden. I det følgende eksempel vises hvordan man kan måle vinklen i forhold til jorden (vaterpas).



## 28.1 Ledningsdiagrammer



## 28.2 Arduino bibliotek

Følgende bibliotek skal installeres: *mpu6050\_tockn*

**MPU6050\_tockn** by tockn

1.5.2 installed

It can get accel, gyro, and angle data. Arduino library for easy communicating with the MPU6050.  
[More info](#)

## 28.3 Software eksempel

```
#include <MPU6050_tockn.h>
#include <Wire.h>

int angleX;
int angleY;

MPU6050 mpu6050(Wire);

void setup() {
    Serial.begin(9600);
    Wire.begin();
    mpu6050.begin();
    mpu6050.calcGyroOffsets(true);
}

void loop() {
    mpu6050.update();

    angleX = mpu6050.getAngleX();
    angleY = mpu6050.getAngleY();

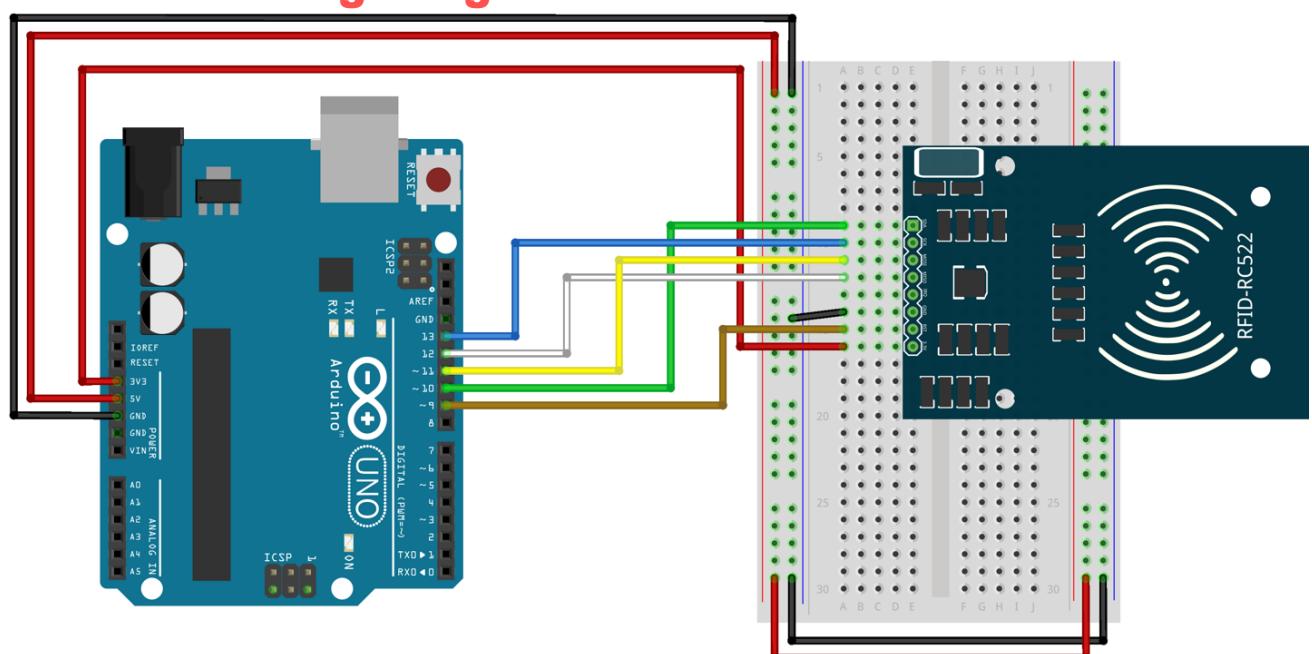
    Serial.print("X: ");
    Serial.print(angleX);
    Serial.print("  ");
    Serial.print("Y: ");
    Serial.print(angleY);
    Serial.println();
}
```

## 29 RFID

RFID-læser, kan læse et RFID-kort. Den kan bla. bruges til et adgangssystem.



### 29.1 Ledningsdiagrammer



## 29.2 Arduino bibliotek

Følgende bibliotek skal installeres: *mfrc522*

[MFRC522](#) by GithubCommunity

1.4.10 installed

Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface. Arduino RFID Library for MFRC522 (SPI)  
[More info](#)

## 29.3 Software eksempel

```
#include <SPI.h>
#include <MFRC522.h>

MFRC522 mfrc522(10, 9);

String uid;

void setup() {
    Serial.begin(9600);
    SPI.begin();
    mfrc522.PCD_Init();
}

void loop() {
    if (checkForCard() == true) {
        readCard();
        Serial.print("UID: ");
        Serial.println(uid );
    }
}

bool checkForCard() {
    if(mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
        return true;
    } else {
        return false;
    }
}

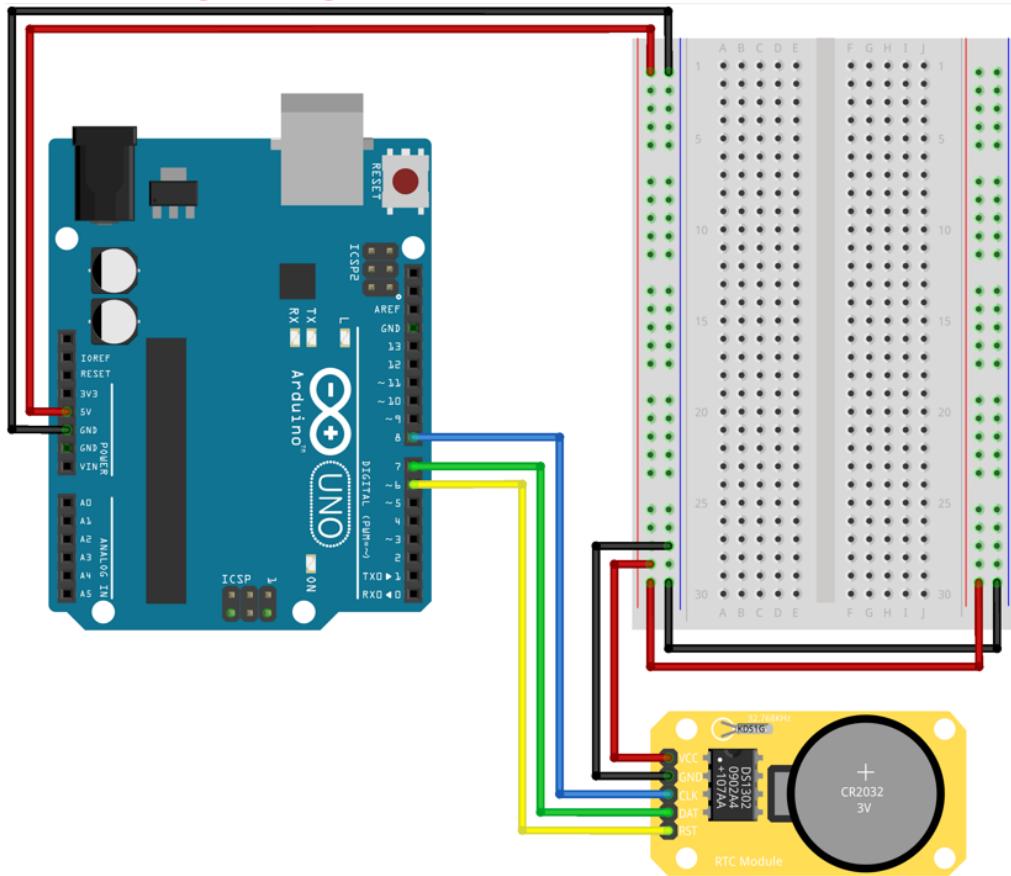
void readCard() {
    uid = "";
    byte readCard[4];
    for ( uint8_t i = 0; i < 4; i++) {
        readCard[i] = mfrc522.uid.uidByte[i];
        uid.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
}
```

## 30 RTC

RTC (Real Time Clock), kan 'huske' tiden, selvom arduinoen har været slukket.



## 30.1 Ledningsdiagrammer

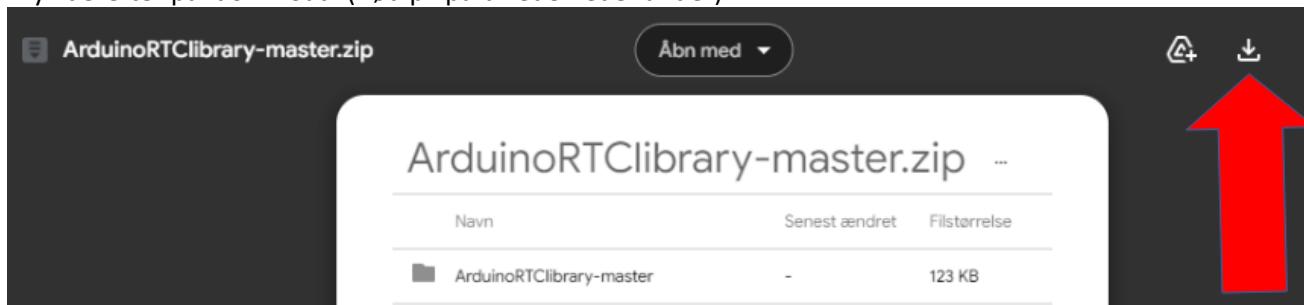


## 30.2 Arduino bibliotek

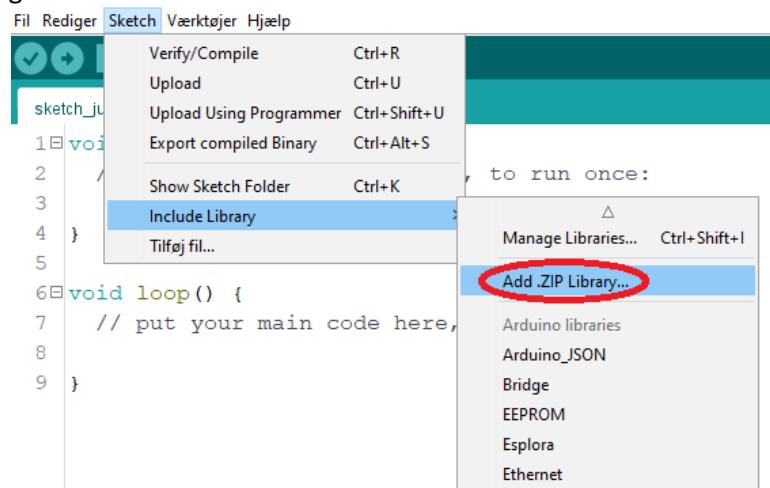
Det Arduino bibliotek som skal installeres, er lidt specielt. Først skal man downloade biblioteket fra nettet. Tryk på følgende link:

<https://drive.google.com/file/d/1GDuLWUANun5toL-XsEaFfaw2tv0mvhjB/view>

Tryk derefter på 'download' (Rød pil på billede nedenunder).



Åben Auduino IDE, vælg Sketch->Include Library->Add .ZIP Library (se billede nedenunder)  
Vælg den fil, som du lige har downloadet overfor.



### 30.3 Software eksempel

```
#include <virtuabotixRTC.h>
virtuabotixRTC myRTC(8, 7, 6);

int second = 0;           // 0 - 59
int minute = 0;          // 0 - 59
int hour = 12;           // 0 - 23
int weekday = 1;          // 0 = mandag, 1 = tirsdag, 2 = onsdag ...
int date = 1;             // 1 - 31
int month = 7;            // 1 - 12
int year = 2023;          // årstal

String weekdayName[7] = {"mandag", "tirsdag", "onsdag", "torsdag", "fredag",
"lordag", "Sondag"};

void setup() {
  Serial.begin(9600);

  //Indstil tiden ved at slette "://" i linjen under.
  //myRTC.setDS1302Time(second, minute, hour, weekday, date, month, year);
}

void loop() {
  myRTC.updateTime();

  Serial.print("Dato: ");
  Serial.print(myRTC.dayofmonth);
  Serial.print("/");
  Serial.print(myRTC.month);
  Serial.print("/");
  Serial.print(myRTC.year);

  Serial.print(" Ugedag: ");
  Serial.print(weekdayName[weekday]);

  Serial.print(" Tid: ");
  Serial.print(myRTC.hours);
  Serial.print(":");
  Serial.print(myRTC.minutes);
  Serial.print(":");
  Serial.print(myRTC.seconds);

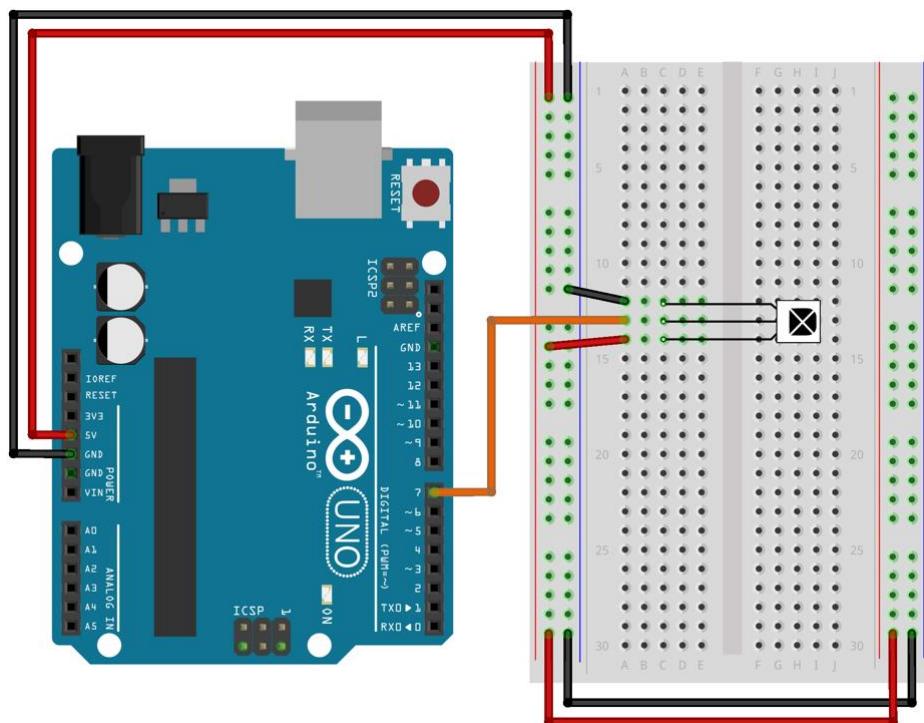
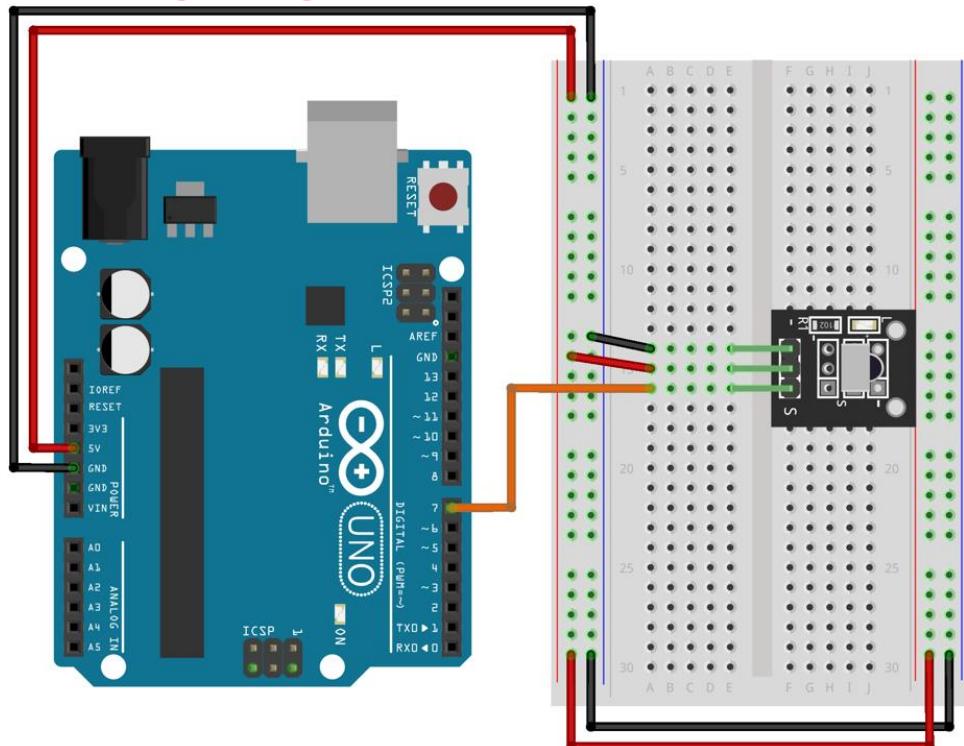
  Serial.println();
}
```

### 31 IR remote

Med en IR remote og en IR modtager kan man 'fjernstyre' ting. Vi har to forskellige modtagere, derfor er der to forskellige ledningsdiagrammer. Koden er den samme for begge modtagere.



## 31.1 Ledningsdiagrammer



## 31.2 Arduino bibliotek

Følgende bibliotek skal installeres: *IRremote*

**IRremote** by shirriff, z3t0, ArminJo

4.1.2 installed

Currently included protocols: Denon / Sharp, JVC, LG / LG2, NEC / Onkyo / Apple, Panasonic / Kaseikyo, RC5, RC6, Samsung, Sony, (Pronto), BangOlufsen, BoseWave, Lego, Whynter, FAST, MagiQuest.  
New: Added FAST Protocol. Changed some function signatures. Improved handling of PULSE\_DISTANCE + PULSE\_WIDTH protocols...  
[More info](#)

### 31.3 Software eksempel

Simpelt program til at læse et tastetryk med.

```
#include <IRremote.h>

#define IR_PIN 7

IRrecv irRecv(IR_PIN);

long irKey;
long lastKey;

void setup() {
  Serial.begin(9600);
  irRecv.enableIRIn();
}

void loop() {
  irKey = readIr();
  if(irKey != 0) {
    Serial.println(irKey, HEX);
  }
}

long readIr() {
  if (irRecv.decode()) {
    irKey = irRecv.decodedIRData.decodedRawData;
    if(irKey == 0) {
      irKey = lastKey;
    }
    lastKey = irKey;
    irRecv.resume();
    return lastKey;
  } else {
    return 0;
  }
}
```

Program til at genkende nogle specifikke taster med.

```
#include <IRremote.h>

#define KEY1 0xBA45FF00 //indsæt den key du vil bruge
#define KEY2 0xB946FF00 //indsæt den key du vil bruge

#define IR_PIN 7

IRrecv irRecv(IR_PIN);

long irKey;
long lastKey;

long lastTime;

void setup() {
  Serial.begin(9600);
  irRecv.enableIRIn();
}

void loop() {
  if(millis() > lastTime + 100) {
    lastTime = millis();
    irKey = readIr();
    if(irKey != 0) {
      switch(irKey) {
        case KEY1:
          Serial.println("du trykker på key1");
          break;
        case KEY2:
          Serial.println("du trykker på key2");
          break;
        default:
          Serial.println("ukendt key");
      }
    } else {
      Serial.println("no input");
    }
  }
}

long readIr() {
  if(irRecv.decode()) {
    irKey = irRecv.decodedIRData.decodedRawData;
    if(irKey == 0) {
      irKey = lastKey;
    }
    lastKey = irKey;
    irRecv.resume();
    return lastKey;
  } else {
    return 0;
  }
}
```

## 32 Bluetooth

Bluetooth er en trådløs protokol til kommunikation over korte afstande – nogle få meter. Der skal bruges to moduler, et som sender og et som modtager. De to moduler skal være parret med hinanden for at kunne snakke sammen.

Bluetooth modulerne ligger parret to og to sammen som vist på billedet og det er vigtigt at tallet er ens på de to moduler man vælger. Der er ingen forskel på de to moduler, og de kan begge sende og modtage.



Når modulerne har kontakt med hinanden, så skal LED'en på begge moduler lave 2 hurtige blink og derefter slukke i ca. 3 sekunder (gentages for evigt).

Derudover skal man bruge to forskellige modstande (2 af hver), som er vigtige for at beskytte modulerne fra at brænde af. Disse modstande fås af instruktørerne og kan ses på nedenstående billeder.

(Man skal også i eksemplet bruge to potentiometre, men dem kan man selv tage).

I eksemplet er der brugt en Arduino Nano, men man kan også bruge en Arduino Uno.



1 kΩ



2 kΩ

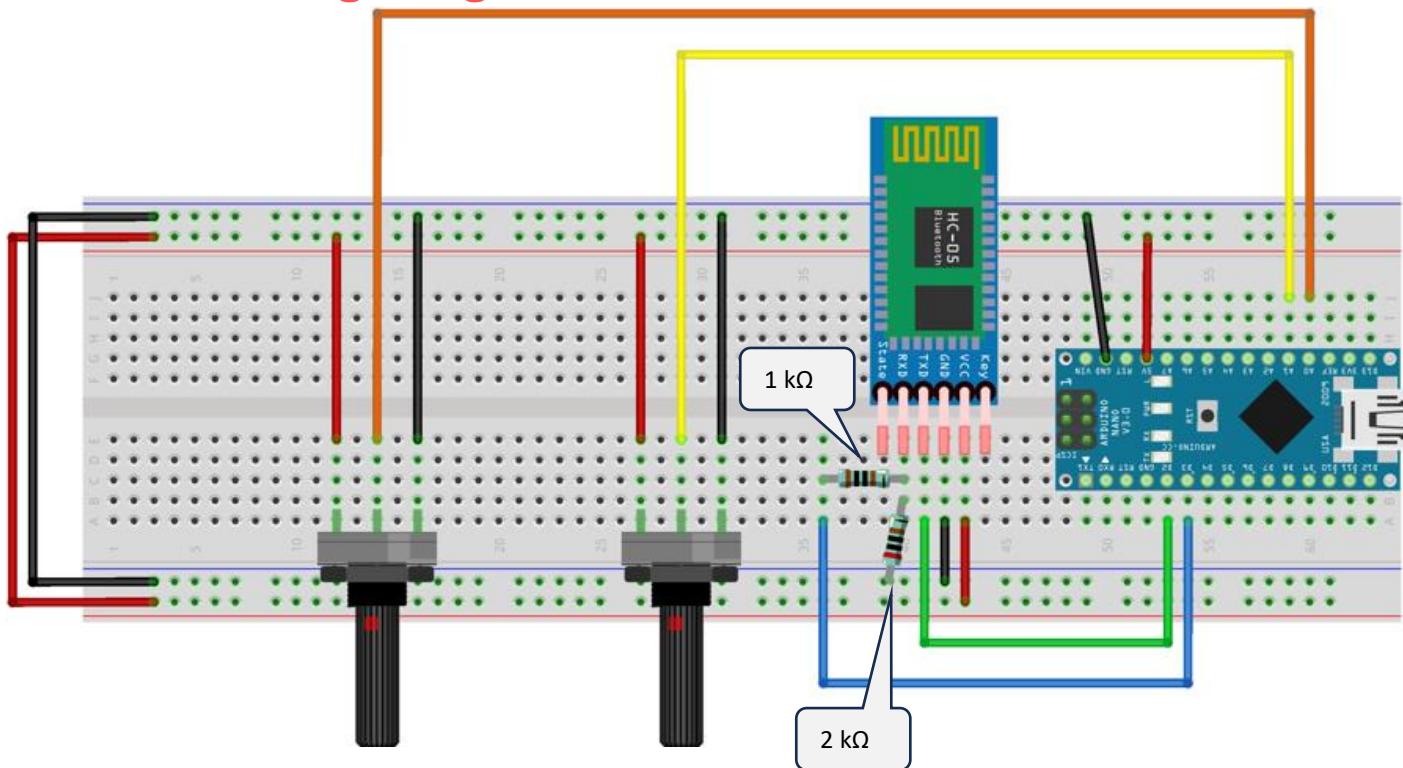


Ingen røde streger.



Én Rød streg forrest.

### 32.1 Ledningsdiagrammer sender



## 32.2 Software eksempel sender

```
#include <SoftwareSerial.h>

#define RX_PIN 2
#define TX_PIN 3
#define X_POT_PIN A0
#define Y_POT_PIN A1

SoftwareSerial BTSerial(RX_PIN, TX_PIN);

int xPotValue;
int yPotValue;

long lastTime;

void setup() {
  Serial.begin(9600);
  BTSerial.begin(9600);
  lastTime = millis();

  pinMode(X_POT_PIN, INPUT);
  pinMode(Y_POT_PIN, INPUT);
}

void loop() {
  xPotValue = analogRead(X_POT_PIN);
  yPotValue = analogRead(Y_POT_PIN);

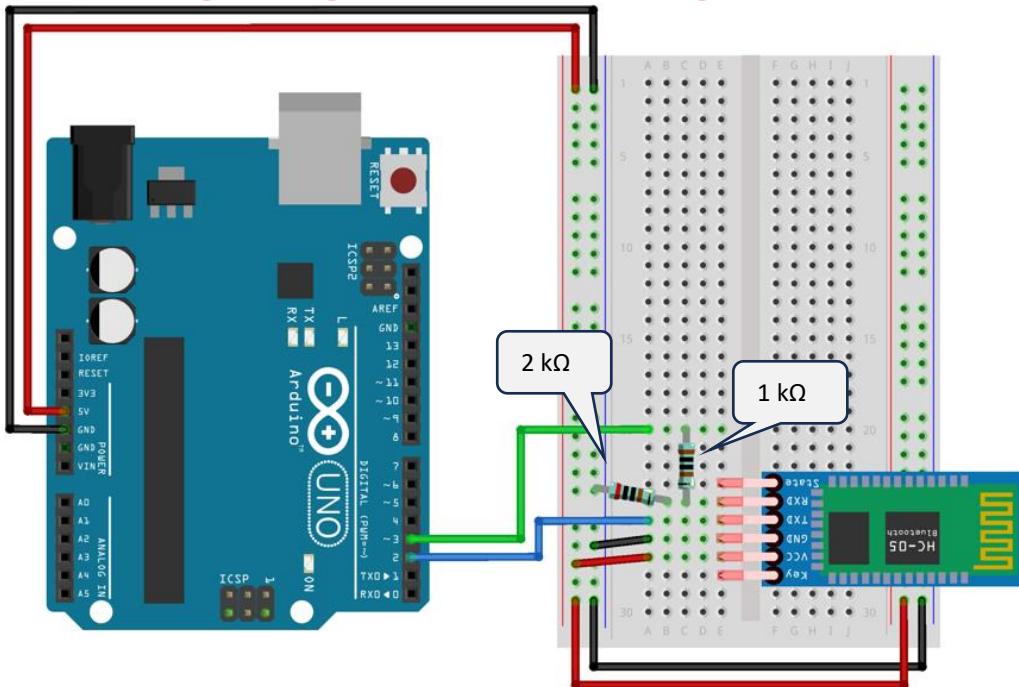
  Serial.print(xPotValue);
  Serial.print(" ");
  Serial.println(yPotValue);

  if(millis() - lastTime > 100) {
    lastTime = millis();

    byte xTransmit = map(xPotValue, 0, 1023, 0,127);      //værdier fra 0-127
    tildeles X
    byte yTransmit = map(yPotValue, 0, 1023, 128, 255);  //værdier fra 128-255
    tildeles Y

    BTSerial.write(xTransmit);
    BTSerial.write(yTransmit);
  }
}
```

### 32.3 Ledningsdiagrammer modtager



### 32.4 Software eksempel modtager

```
#include <SoftwareSerial.h>

#define RX_PIN 2
#define TX_PIN 3

int xValue;
int yValue;

SoftwareSerial BTSerial(RX_PIN, TX_PIN);

void setup() {
  Serial.begin(9600);
  BTSerial.begin(9600);
}

void loop() {
  byte data;
  if(BTSerial.available()) {
    data = BTSerial.read(); //incomming data fra bluetooth
    if(data < 128) {
      xValue = data; //værdier fra 0-127 tildeles X
    } else {
      yValue = data-128; //værdier fra 128-255 tildeles Y
    }
  }

  Serial.print(xValue);
  Serial.print(" ");
  Serial.print(yValue);
  Serial.println();
}
```

# 33 C/C++ Programmeringsguide

Dette er en lille guide til hvordan man programmerer i C/C++. Det er kun en lille del af C/C++ som gennemgås, men den skulle gerne dække de fleste af de metoder du får brug for i forløbet.

De fleste linjer i C afsluttes med ";"(semikolon, sidder over , på tastaturet)

## 33.1 Variable

Variable bruges til at gemme værdier imens programmet kører. Det er vigtigt at man vælger den rigtige variable type, da koden ellers ikke vil virke efter hensigten. Her er nogle af de typer som C har.

Type	Beskrivelse	Værdier
<b>int</b>	Heltal, (integer)	Fx -3, -2, -1, 0, 1, 2, 3
<b>long</b>	Stort heltal (integer)	Fx 3276474
<b>float</b>	Kommatal, (floating point) HUSK: 'kommaet' skal være et punktum!	-1.23, 4.3
<b>bool</b>	Boolsk værdi (to værdier)	true, false
<b>char</b>	Bogstav (character)	'A', 'b', 'q'

Her er nogle eksempler på, hvordan variabler kan erklæres (oprettes).

```
int i;  
long k = 5;  
  
float f;  
float p = -1.23;
```

```
bool b;  
bool a = true;  
  
char c;  
char d = 'A';
```

Der er flere måder at bruge sine variable på i koden. Her er et par eksempler:

Her laver man en simpel udregning og gemmer resultatet i "a".

```
a = 5 * b;
```

Disse to linjer laver det samme! De lægger én til "a".

```
a = a + 1;  
a++;
```

## 33.2 Betingelser

Hvis man skal træffe et valg mellem to muligheder fx om "a" > 5, så skal man bruge en **if** sætning.

Her er et eksempel på en **if** sætning:

```
if (a > 5) {  
    // Det er sandt  
} else {  
    // Det er falsk  
}
```

Hvis "a" er større end 5, så bliver linjen med 'Det er sandt' udført. Hvis det er falsk, så bliver linjen med 'Det er falsk' udført.

Husk altid parenteser () rundt om betingelsen. Hvis man ikke har brug for else sætningen, kan den udelades.

Tabel over nogle af de betingelser man kan bruge.

Betingelse	Beskrivelse	Eksempel
==	Lig med, er to værdier ens (ja, det skal være ==)	a == 5
!=	Ikke lig med, er to værdier forskellige	a != 10
>	Større end	a > 4
>=	Større end eller lig med	a >= 4
<	Mindre end	a < -3
<=	Mindre end eller lig med	a <= 0

### 33.3 Loops

Hvis den samme kode skal køre flere gang, så kan man bruge et loop (gentagelse). C indeholder flere måde at lave loops på, men her vil der kun blive beskrevet en.

for loop'en bruges når den samme kode skal køre flere gange. Her er et eksempel på et for loop:

```
int i;
for (i=0; i<10; i++) {
    // Denne kode kører 10 gange
}
```

Denne loop udfører den samme linje (blok) 10 gange.

De 'krøllede' {} parenteser laver en blok, dvs. at alle de linjer som er inde i mellem {} hænger sammen.

### 33.4 Funktioner

En funktion er et modul som løser et mindre problem. På den måde kan man dele sin kode op i mindre dele så den bliver lettere at forstå og genbruge.

Denne funktion kan bruges til at få en bil til at køre frem.

```
void driveForward() {
}
```

Man kan også lave en funktion som læser fra en sensor og returnerer resultatet.

```
int readSensor() {
}
```

Det var bare en kort beskrivelse af hvad man kan bruge funktioner til. De har mange flere muligheder med argumenter mv. som ikke er vist her.

## 34 Arduino bibliotek

Arduino har mange funktioner som man kan bruge i sin kode. De er alle fint beskrevet på følgende side:  
<https://www.arduino.cc/reference/en/>.

Herunder er nogle få af de funktioner vi bruger mest, beskrevet.

### 34.1 Styring af digitale og analoge pins

Her er nogle funktioner som bruges når man skal 'snakke' med de komponenter som man har sat til sit Arduino board.

Funktion	Beskrivelse	Eksempel
<b>pinMode()</b>	Indstiller om den pin man angiver skal være input (styres udefra) eller output (styrer noget) INPUT, OUTPUT, INPUT_PULLUP	<code>pinMode(LED_PIN, OUTPUT); pinMode(BUTTON_PIN, INPUT);</code>
<b>digitalWrite</b>	Bruges til at tænde eller slukke for signalet på en pin, som er sat til OUTPUT.	<code>digitalWrite(LED_PIN, HIGH);</code>
<b>digitalRead</b>	Bruges til at læse signalet som kommer ind fra en pin, som er sat til INPUT.	<code>a = digitalRead(BUTTON_PIN);</code>
<b>analogWrite</b>	Skriv en analog værdi. Værdien skal være 0-255 og svarer til 0-5V. Det er kun pins <b>3, 5, 6, 9, 10, 11</b> som kan bruges.	<code>analogWrite(MOTOR_PIN, 500);</code>
<b>analogRead</b>	Bruges til at læse en analog værdi med. Resultatet er 0-1023 og svarer til 0-5V. Det er kun pins <b>A0-A5</b> som kan bruges. (+A6-A7 på Nano)	<code>k = analogRead(LDR_PIN);</code>

## 34.2 Delay

CPU'en kører meget hurtigt, og man har derfor tit brug for at få koden til at holde en lille pause. Funktionen `delay()` holder en pause i millisekunder (tusindedele af et sekund, 1000 millisekunder = 1 sekund).

Følgende linje laver en pause på 500 millisekunder (0,5 sekunder).

```
delay(500);
```

## 34.3 Print

Når man skal debugge sin kode, kan det være en god ide at printe noget ud på skærmen (serial monitor). Inden man kan printe, skal man åbne forbindelsen og det gøres på følgende måde:

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

Derefter kan man printe på følgende måde:

```
Serial.print("Hej ");  
Serial.println(v);
```

Den første linje printer 'Hej' ud, og den anden linje printer værdien af v ud og laver en ny linje.

# 35 Arduino Uno vs Arduino Nano

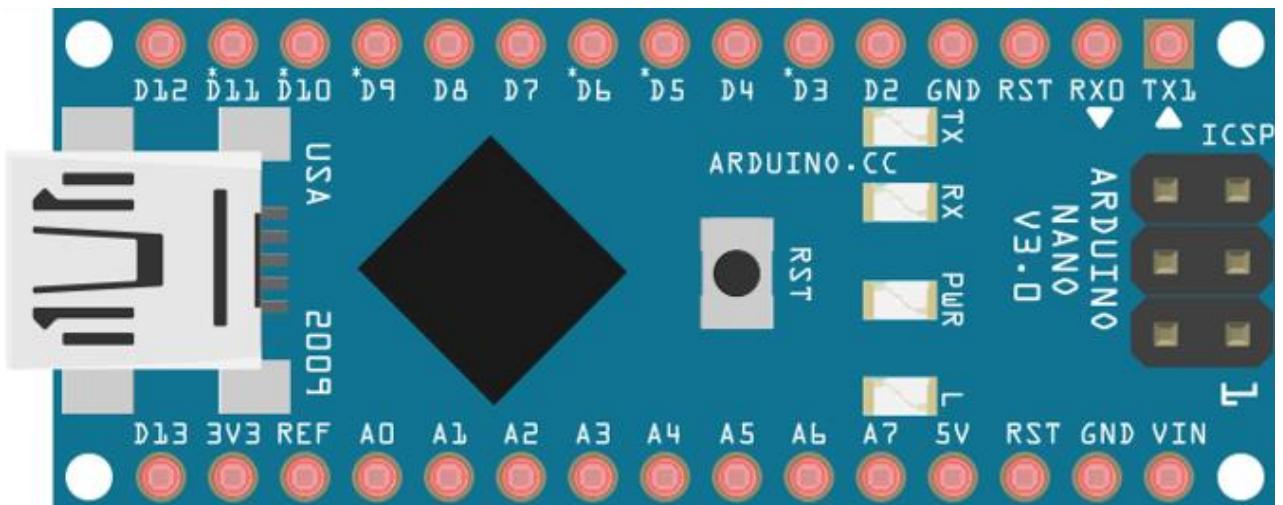
Vi har to forskellig Arduino boards:

- Uno (stort)
- Nano (lille)

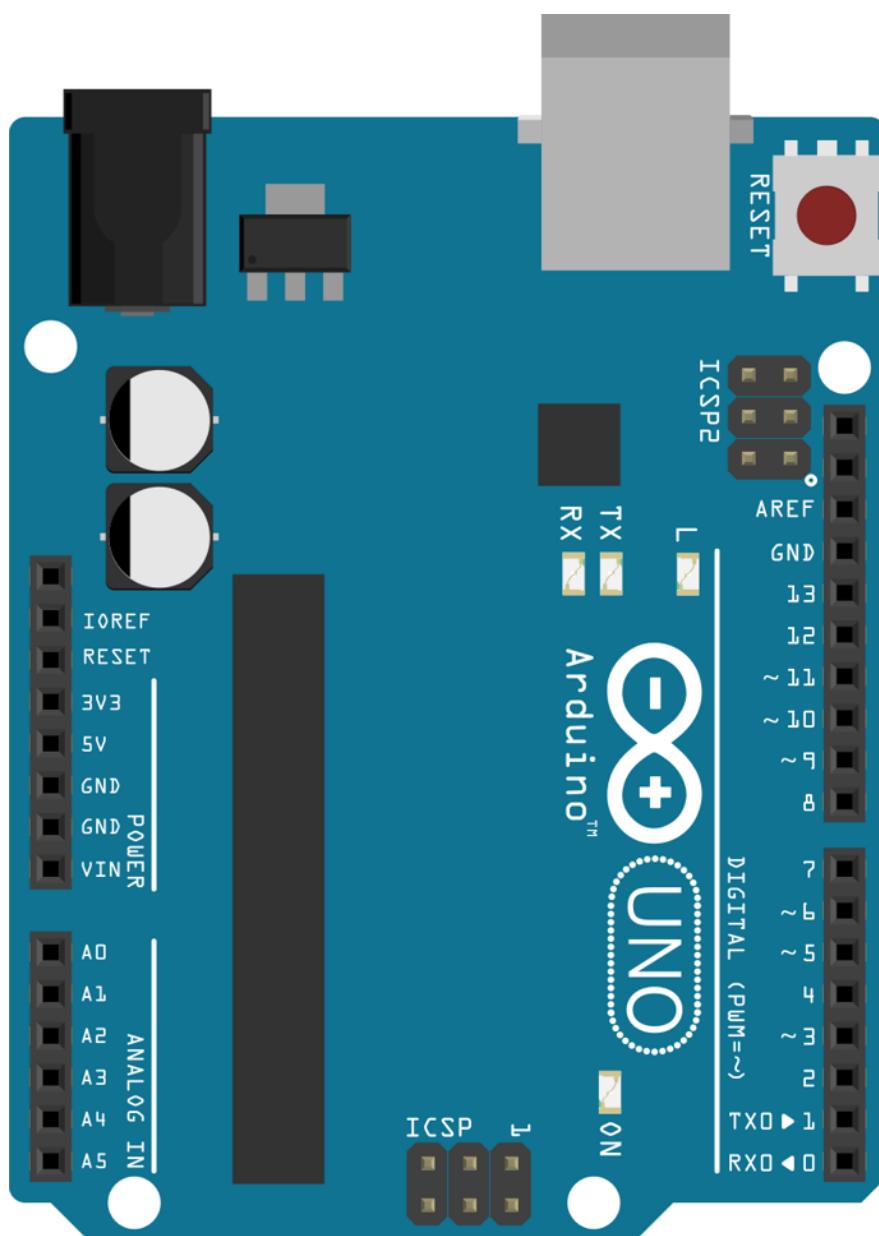
Alle på campen starter med et Uno board. Nano boardet er meget mindre, men har faktisk flere IO pins end Uno! Det er den samme CPU som sidder på begge board, så de kan præcis det samme og kører lige hurtigt. Her er lidt fælles data for de to boards og for sjov sammenlignet med en PC (2021).

Enhed	Type	PC (gange hurtigere)
<b>CPU</b>	ATMEGA328	Intel/AMD
<b>Flash (kode)</b>	32 kbyte	SSD, 1 TB (x30.000.000)
<b>RAM</b>	2 kbyte	8 Gbyte (x4.000.000)
<b>EEPROM</b>	1 kbyte	SSD
<b>Clock</b>	16 MHz	4 GHz/core (x250/core)
<b>Hastighed</b>	16 MIPS	400.000 MIPS (x25.000)

Her er et billede af hver for sammenligning.



Nano'en har to ekstra pins, som hedder A6 og A7. Disse to pins kan kun bruges som analoge pin.

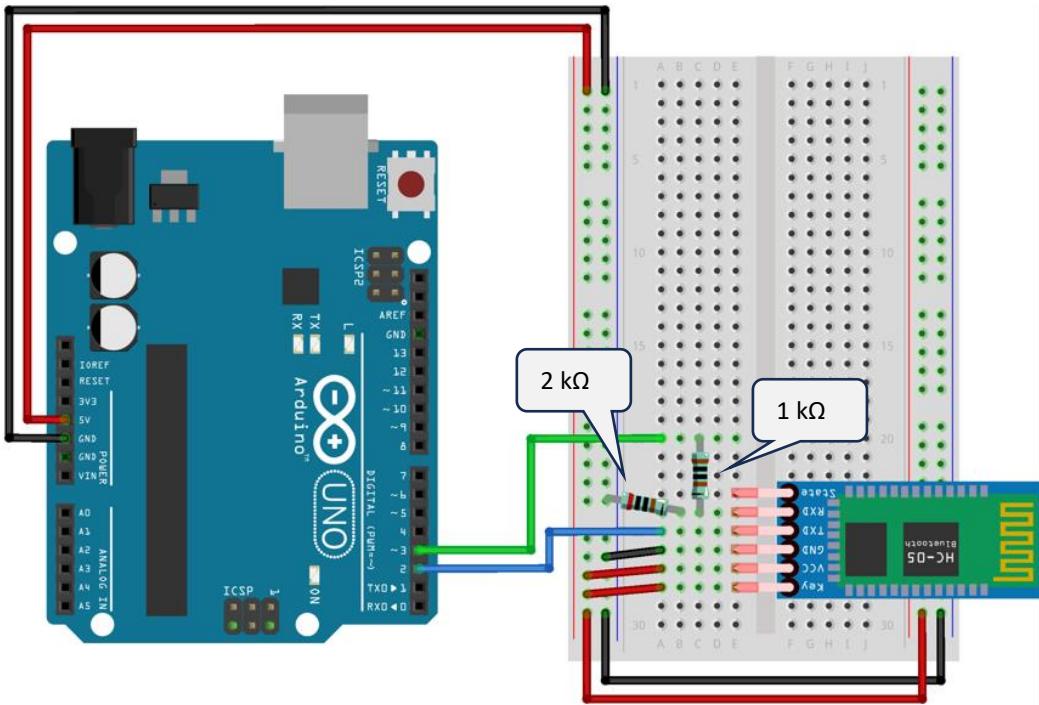


## 36 Bluetooth parring

Dette skal KUN gøres af instruktørerne!

Her er en vejledning i hvordan man kan parrer to bluetooth moduler, så de er klart til brug.

## 36.1 Ledningsdiagrammer



## 36.2 Software setup

```
#include <SoftwareSerial.h>

#define RX_PIN 2
#define TX_PIN 3

SoftwareSerial BTSerial(RX_PIN, TX_PIN);

char dataByte;

void setup() {
  Serial.begin(9600);
  pinMode(RX_PIN, INPUT);
  pinMode(TX_PIN, OUTPUT);

  BTSerial.begin(38400);
  Serial.println("Start UNO");
}

void loop() {

  while(BTSerial.available()) {
    dataByte = BTSerial.read();
    Serial.write(dataByte);
  }

  while(Serial.available()) {
    dataByte = Serial.read();
    BTSerial.write(dataByte);
  }
}
```

## 36.3 Setup

1.

Forbind bluetooth-modulet som vist i ledningsdiagrammet ovenfor.

Vær opmærksom på de to modstande:

2K ohm fra (HC-05)RX til (UNO) GND

1K ohm fra (HC-05)RX til (UNO) D4

2.

Uplode koden "HC\_05\_AT\_mode\_setup\_kode" til arduino UNO (softwaren ovenfor)

3.

Kontroller at LED'en på HC-05 blinker langsomt (0.25Hz)

Hvis den derimod blinker hurtigt:

Kontroller at ledningen fra KEY/EN på HC-05 er forbundet til 5V. Tag USB stikket ud af arduinoen og set det igen. Det er ikke nok at trykket reset på arduinoen.

4.

Åben serial monitoren

5.

Kontroller at serial monitor er sat til "9600 baud" og "Both NL & CR"

Det er vigtigt at den IKKE er sat til "No line ending" som ellers er standart.

6.

Tjek at der hul igennem til HC-05 ved at skrive "AT" i serial monitor. Tryk enter for at sende beskeden.

Hvis der er hul igennem, skal HC-05 svare til din serial monitor med "OK".

!!! Det kan være forskelligt om HC-05 moduler er indstillet til 9600 eller 38400 fra fabrikken.

Hvis det ikke lykkedes at få hul igennem til HC-05, så prøv at skifte mellem de to værdier i koden og i Serial monitor.

Nogen HC-05 kan også kræver at baud raten skal være sat til 19200 igennem serial terminalen, mens BTSerial er sat til 38400.

## 36.4 Opsætning af HC-05 som SLAVE

HC-05 er som standart indstillet til at være i SLAVE mode.

Du kan tjekke om dette er tilfældet ved at sende AT kommandoen: "AT+ROLE?"

Hvis HC-05 svarer med "+ROLE:0" betyder det at den er indstillet til SLAVE mode

Kontroller HC-05 baudrate ved at sende AT kommandoen "AT+UART?"

Hvis du modtager 38400, skal den skiftes til 9600. Det gøres ved at sende AT kommandoen "AT+UART=9600,0,0"

Få adressen på det aktuelle HC-05 modul ved at sende AT kommandoen "AT+ADDR?"

HC-05 svarer med noget lignende: "+ADDR:98D3:51:F603B6"

Gem adressen til senere. I dette tilfælde: 98D3:51:F603B6

Hurtig opsætning af SLAVE:

Send "AT", modtag "OK"

Send "AT+UART?", modtag "9600" eller "38400"

Hvis det er nødvendigt: Send "AT+UART=9600,0,0"

Send "AT+ADDR?", modtag adresse

Opsætning af SLAVE modul er færdigt

## 36.5 Opsætning af HC-05 som MASTER

Kontroller HC-05 baudrate ved at sende AT kommandoen "AT+UART?"

Hvis du modtager 38400, skal den skiftes til 9600. Det gøres ved at sende AT kommandoen "AT+UART=9600,0,0"

Indstil HC-05 modulet til MASTER mode ved at sende AT kommandoen "AT+ROLE=1"

Indstil HC-05 modulet til at kun at forbinde til én specifik adresse ved at sende AT kommandoen "AT+CMODE=0"

Indstil HC-05 modulet til at forbinde til SLAVE adressen med AT kommandoen "AT+BIND=(adresse)"

VIGTIGT! brug komma (,) i stedet for kolon (:) når adressen indstilles

Eksempel: "AT+BIND=98D3,51,F603B6"

### Hurtig opsætning af MASTER:

Send "AT", modtag "OK"

Send "AT+UART?", modtag "9600" eller 38400"

Hvis det er nødvendigt: Send "AT+UART=9600,0,0"

Send "AT+ROLE=1", modtag "OK"

Send "AT+CMODE=0", modtag "OK"

Send "AT+BIND=(SLAVE\_adresse)", modtag "OK"

Opsætning af MASTER module er færdigt

## 36.6 Tjek forbindelse

Sæt strøm til begge moduler efter de er blevet indstillet.

Husk at KEY/EN pin på HC-05 IKKE må være forbundet til 5V længere.

HC-05 modulerne vil blinke hurtigt. Inden for 5 sekunder skal modulerne gerne være forbundet.

Det kan ses at modulerne er forbundet når LED'en på begge moduler laver 2 hurtige blink og derefter slukker i ca. 3 sekunder. (Gentages for evigt)



VIGTIGT! Når begge moduler er forbundet, skrives der på bagsiden af MASTER modulet med sprittusch, "M" samt et vilkårligt tal.

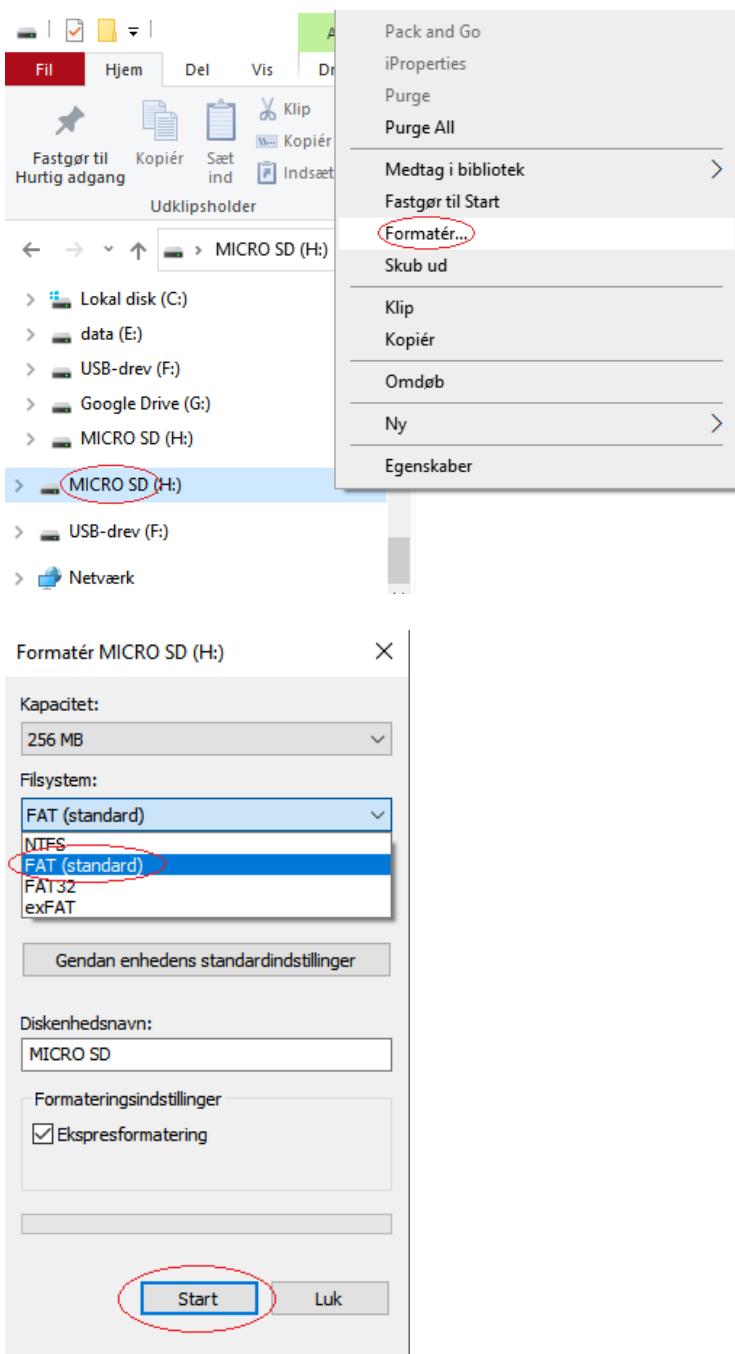
På SLAVE modulet skrives "S" samt samme tal som på master.

Således kan det altid ses hvilke bluetooth moduler der allerede er parret samt hvilken der er MASTER og SLAVE.

## 37 Formater SD kort og overfør filer

Hvis et SD kort ikke er formateret, så gøres det på følgende måde.

Indsæt SD kortet i PC'en og følg billederne nedenfor.



Overfør tre MP3 filer til kortet.

MP3 filerne som er brugt i eksemplet til DF mini player, kan downloades via linket her:

[https://drive.google.com/drive/folders/1Fk7EcSs3k0LDAawjWcV7z5mJYnPnXdQtb?usp=drive\\_link](https://drive.google.com/drive/folders/1Fk7EcSs3k0LDAawjWcV7z5mJYnPnXdQtb?usp=drive_link)

Andet gratis mp3 musik kan findes her:

<https://pixabay.com/music/search/theme/background%20music/>

## 38 Bruges til kopi

### 38.1 Ledningsdiagrammer

### 38.2 Arduino bibliotek

Følgende bibliotek skal installeres:

### **38.3 Software eksempler**