

# KOM GODT I GANG MED ARDUINO!

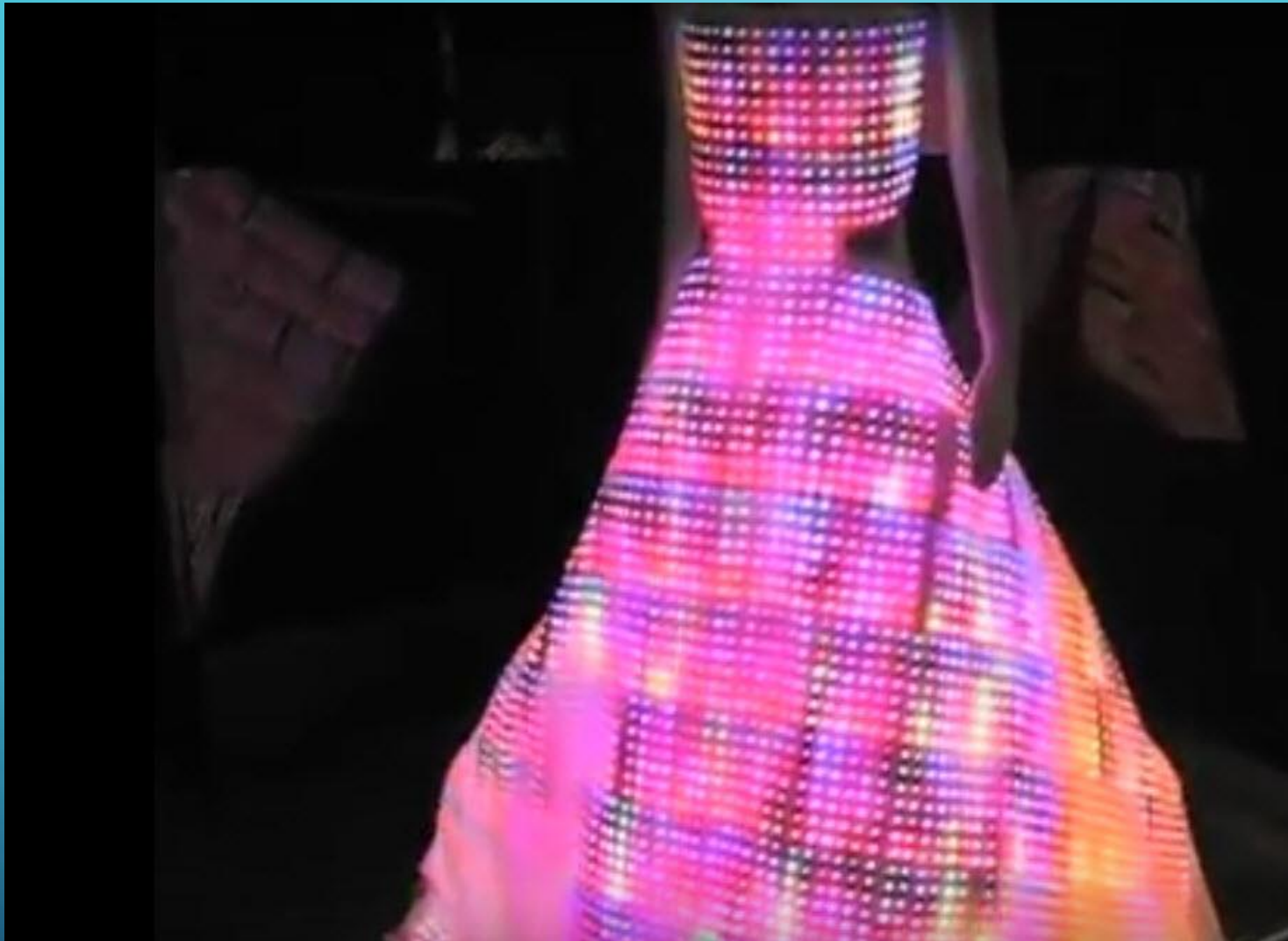


ODENSE DEN 14. NOVEMBER 2016



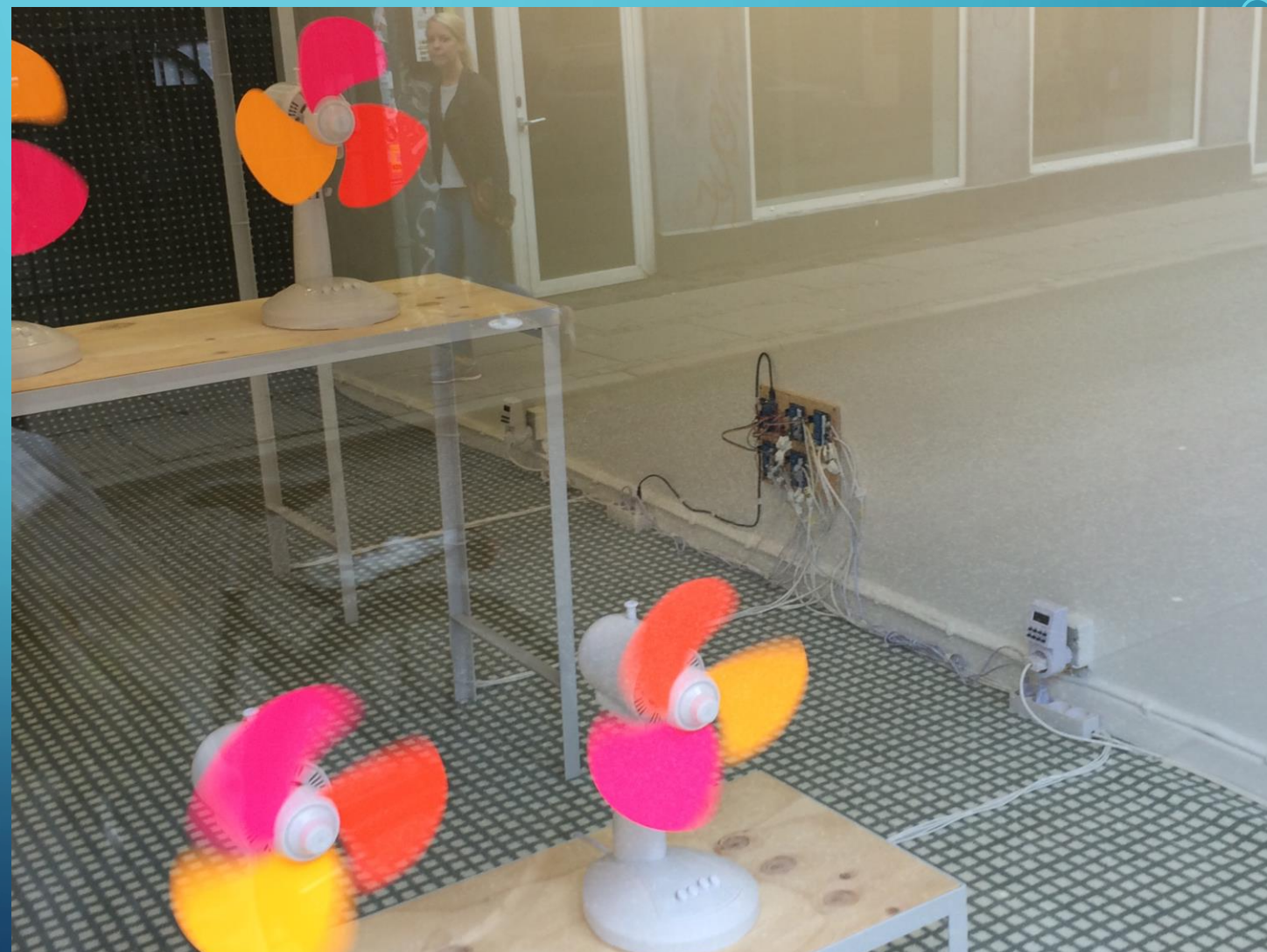
Rubik





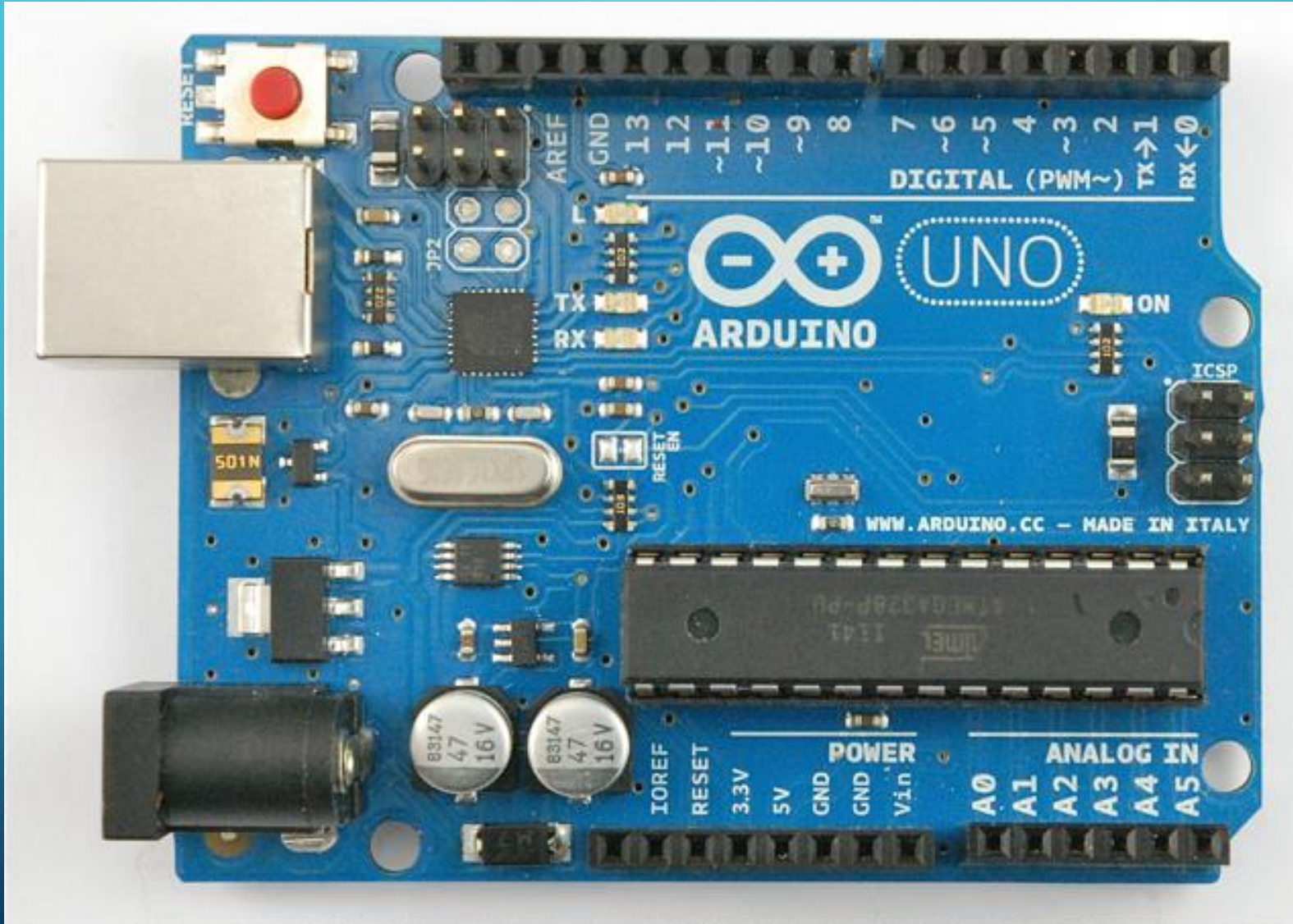
Dress

# 20 MØLLER

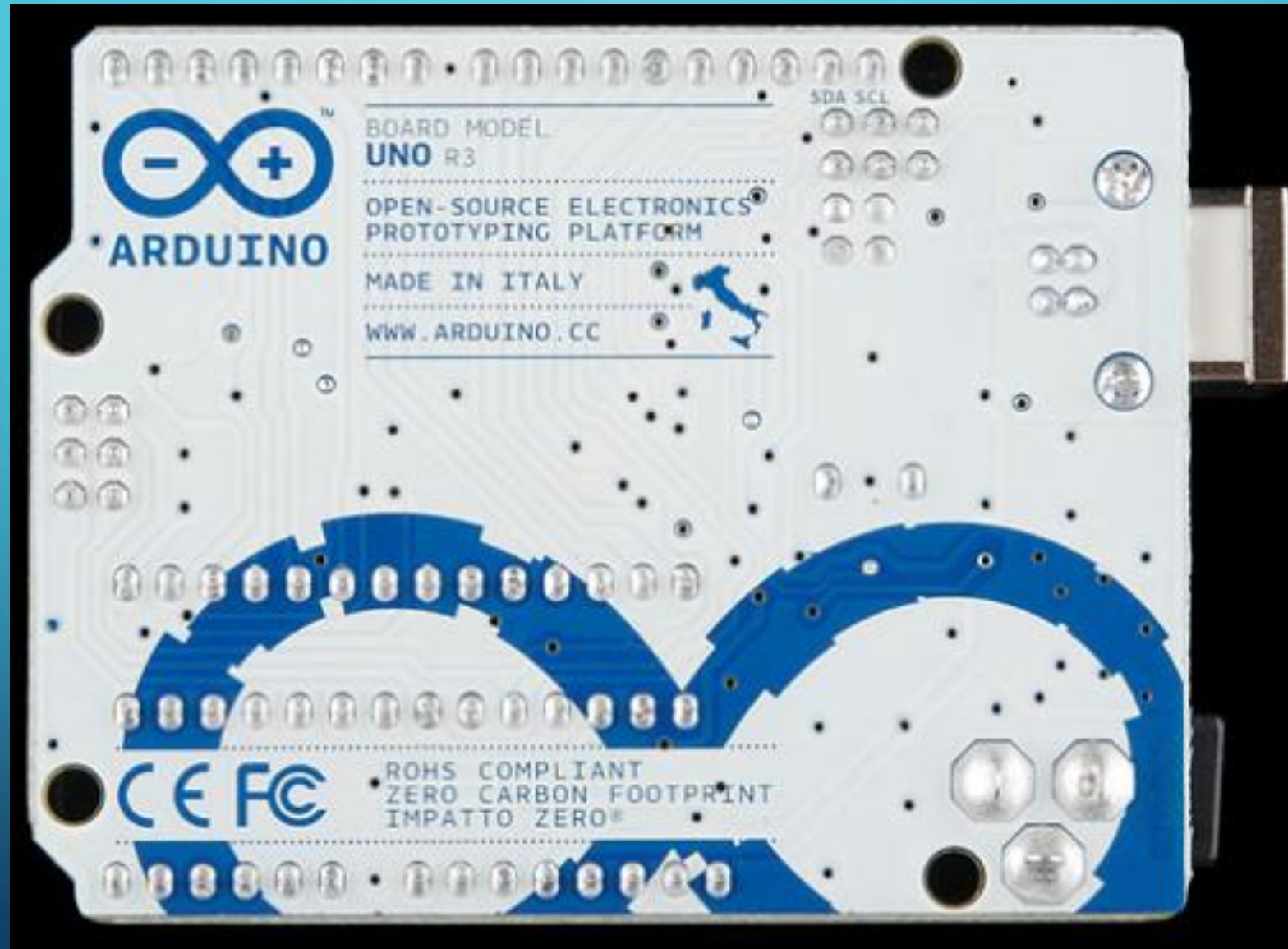




# ARDUINO UNO

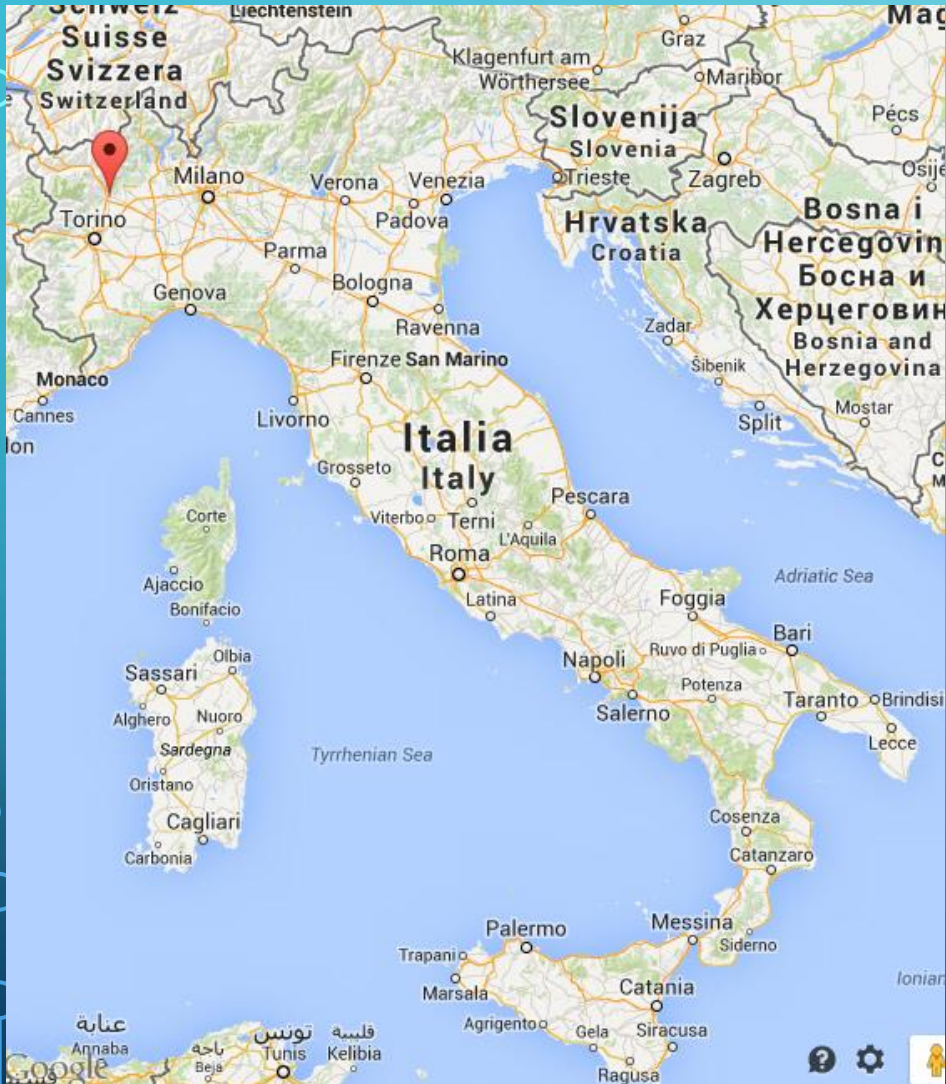


# ARDUINO UNO - BAGSIDEN



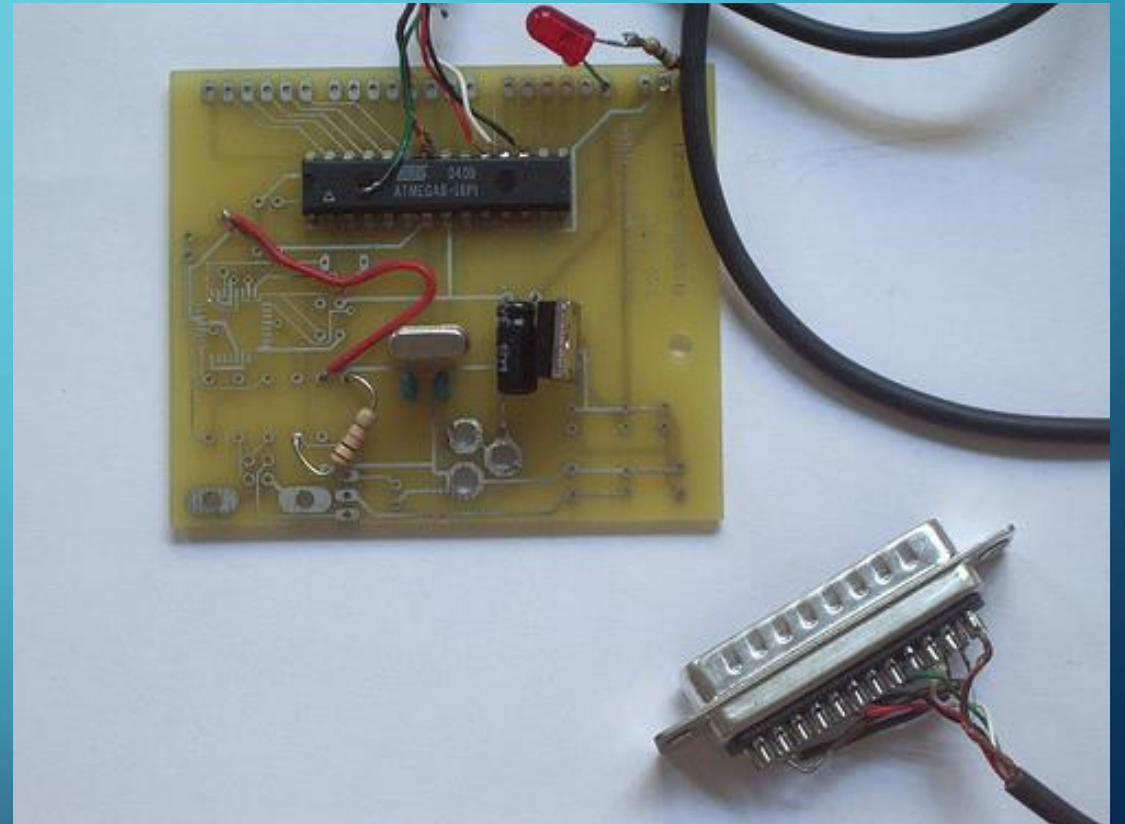
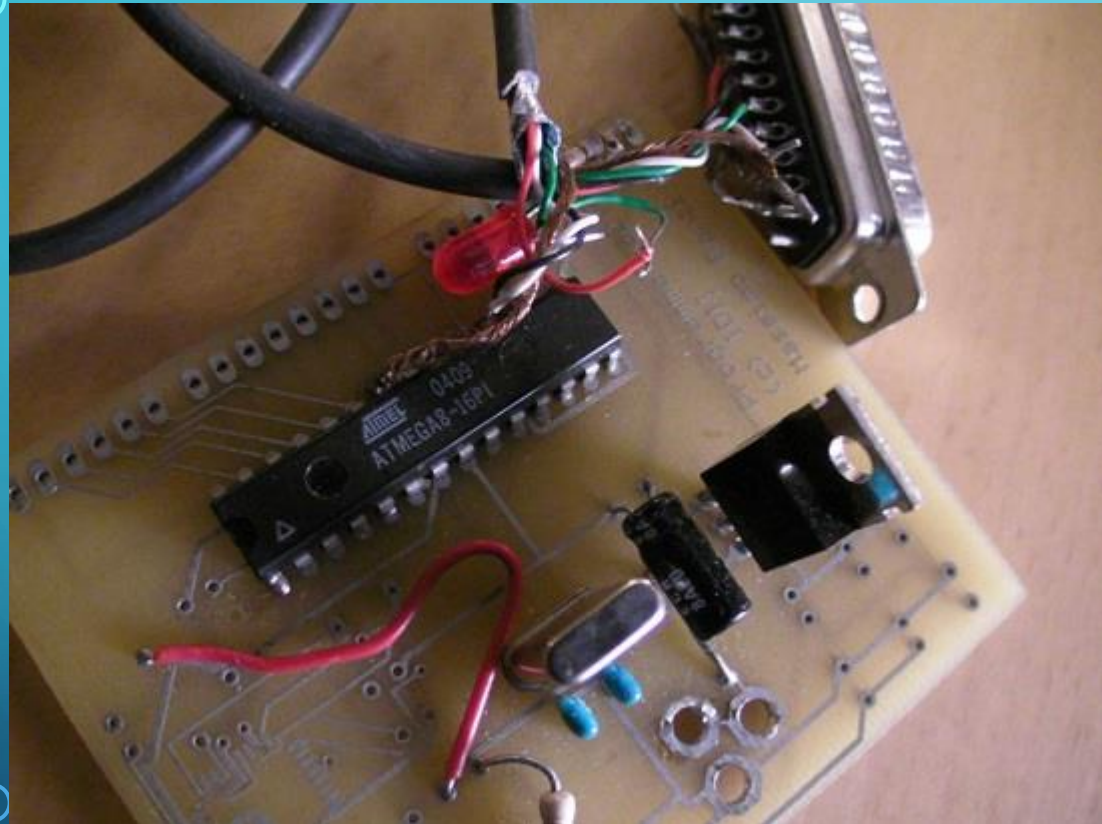


# HISTORIEN OM ARDUINO



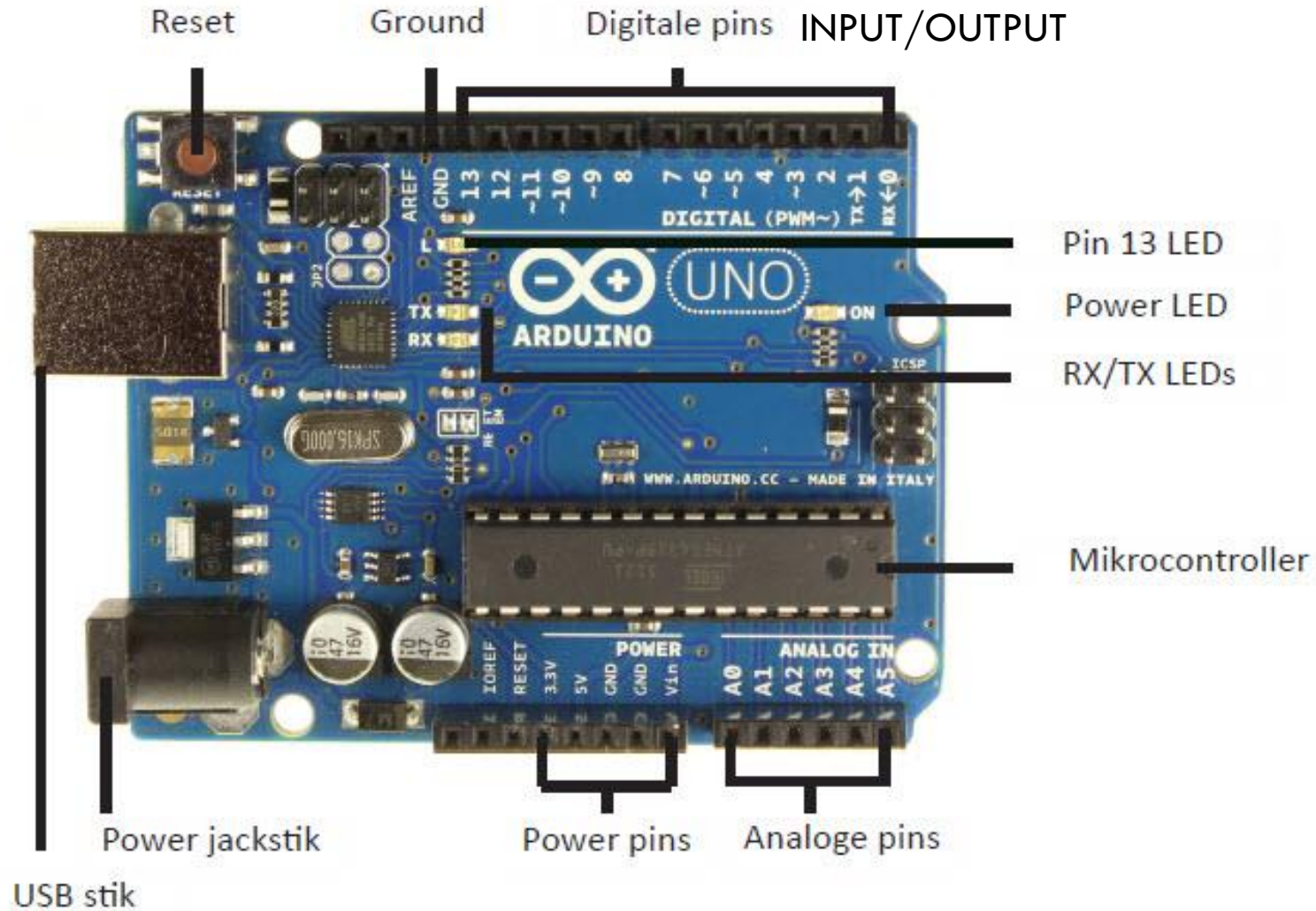


# DEN FØRSTE ARDUINO





# Arduino UNO boardet



De 14 digitale (INPUT/OUTPUT) pins kan programmeres til at være enten en indgang eller en udgang.

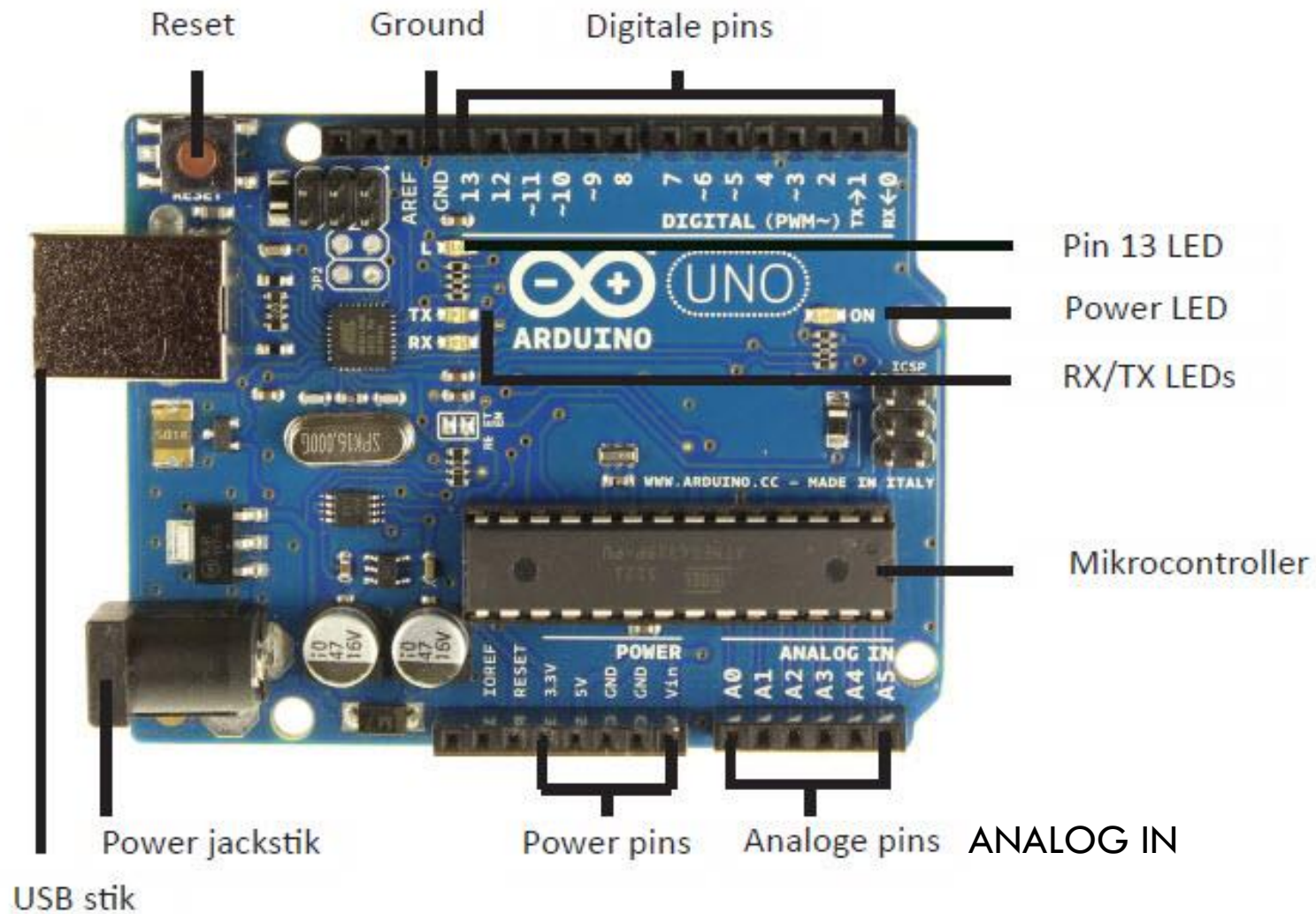
```
pinMode (13, OUTPUT)
```

```
digitalWrite(13, HIGH)
```

```
digitalWrite(13, LOW)
```



# Arduino UNO boardet



Arduino UNO boardet har 6 analoge indgange (A0 – A5)

Boardet indeholder en 10 bit analog til digital konverter. Det betyder at en pin kan måle inputspændinger mellem 0 og 5 volt i heltalsværdier mellem 0 og 1023.

`analogRead(A0)`

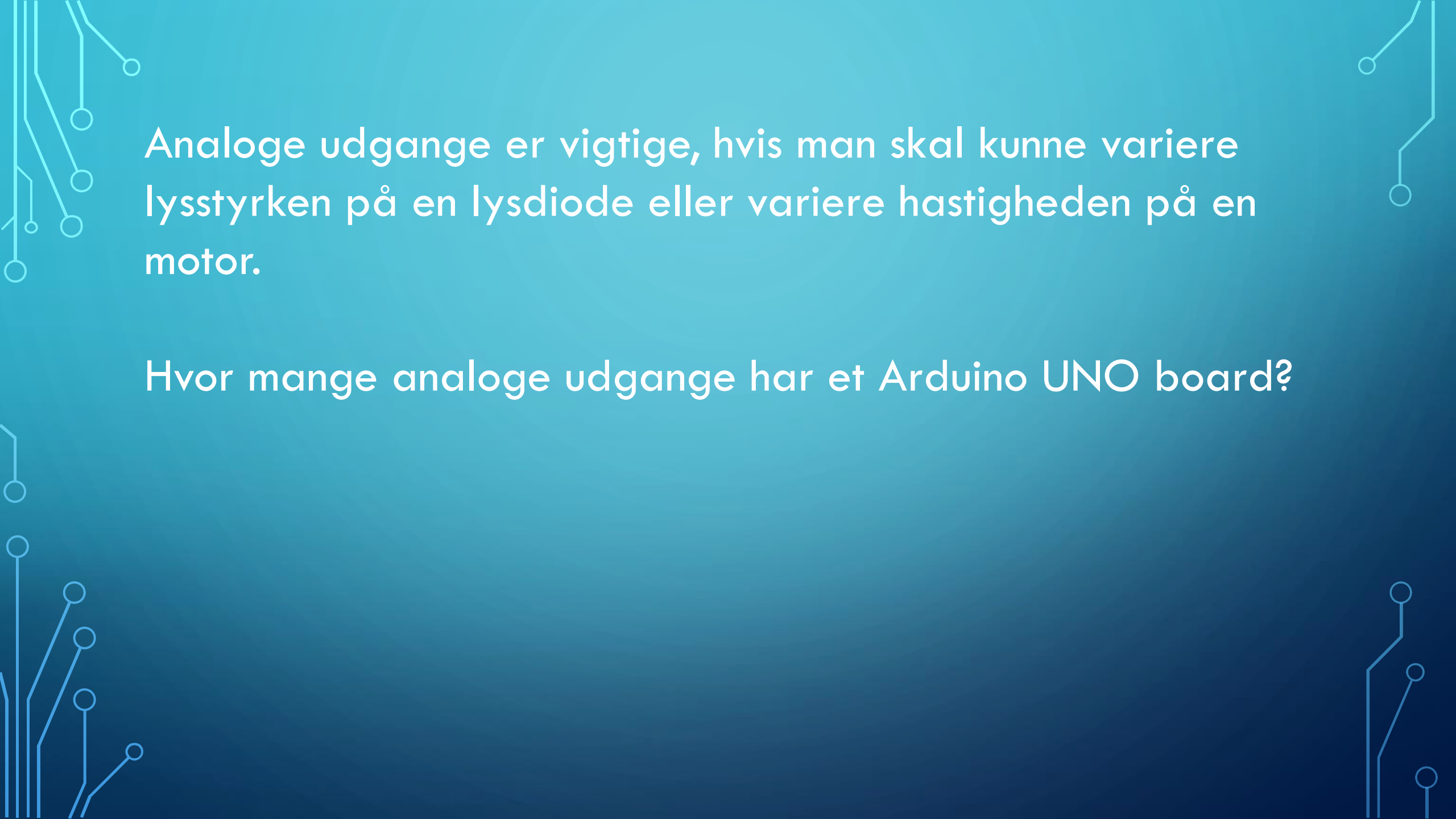
Projekt 3A – Input fra et potentiometer (tal)

Projekt 3B – Input fra et potentiometer (volt)

Projekt 3C – Ændre frekvensen af en lysdiode ved hjælp af et potentiometer

Projekt 4A – Input fra en lysfølsom resistor - LDR



The background is a dark teal gradient. In the corners, there are decorative white line-art patterns resembling circuit board traces and nodes. These patterns are most prominent in the top-left, top-right, and bottom-left corners, with some extending into the bottom-right corner.

Analoge udgange er vigtige, hvis man skal kunne variere lysstyrken på en lysdiode eller variere hastigheden på en motor.

Hvor mange analoge udgange har et Arduino UNO board?

The background is a dark teal gradient. In the corners, there are decorative white line-art patterns resembling circuit traces or a stylized tree structure. These patterns consist of thin lines that branch out and terminate in small circles, mimicking the look of a printed circuit board or a network diagram.

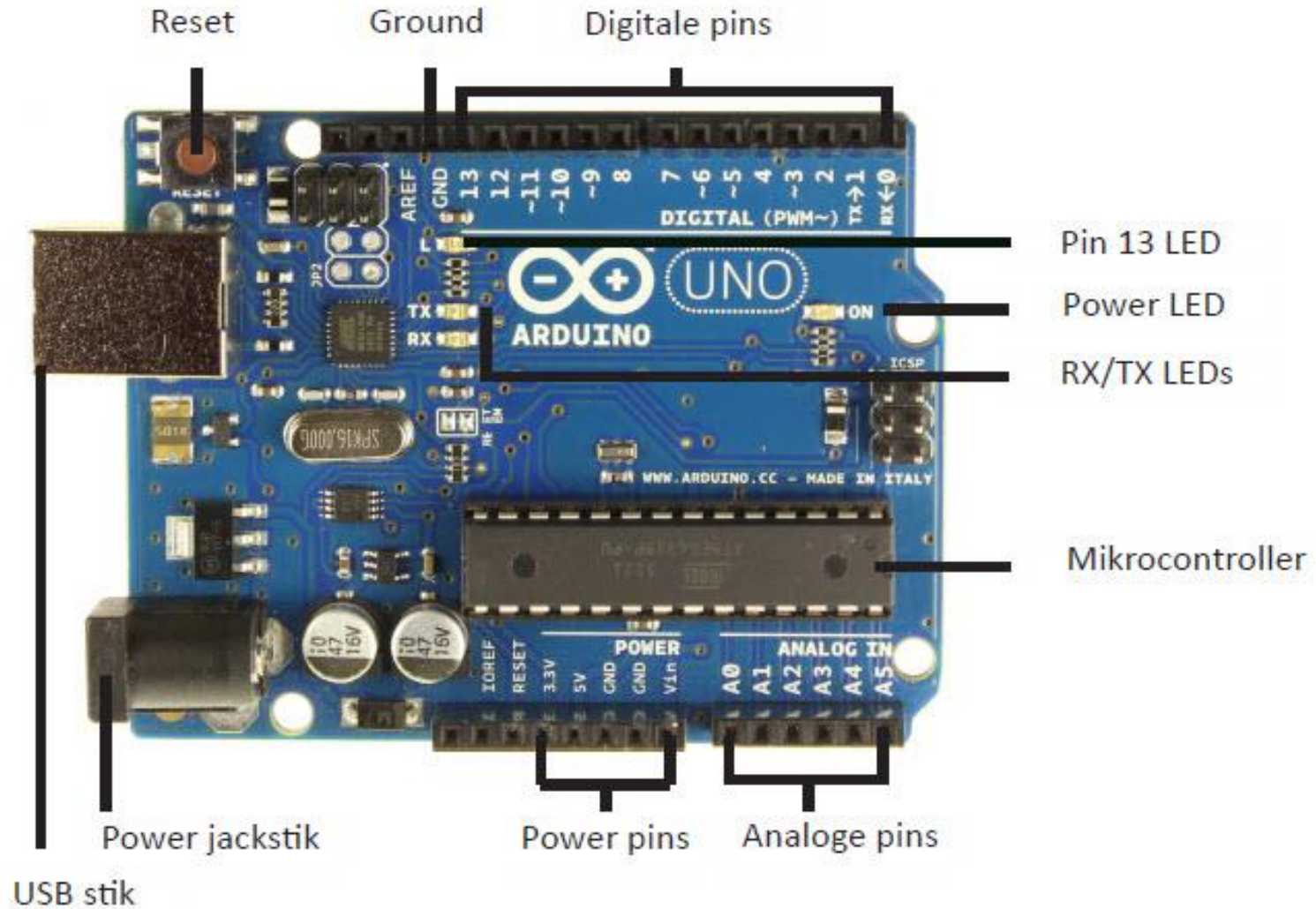
Ingen!

Men man kan generere en analog udgang (OUTPUT) ved hjælp af et trick, som kaldes Pulse-Width-Modulation (PWM)

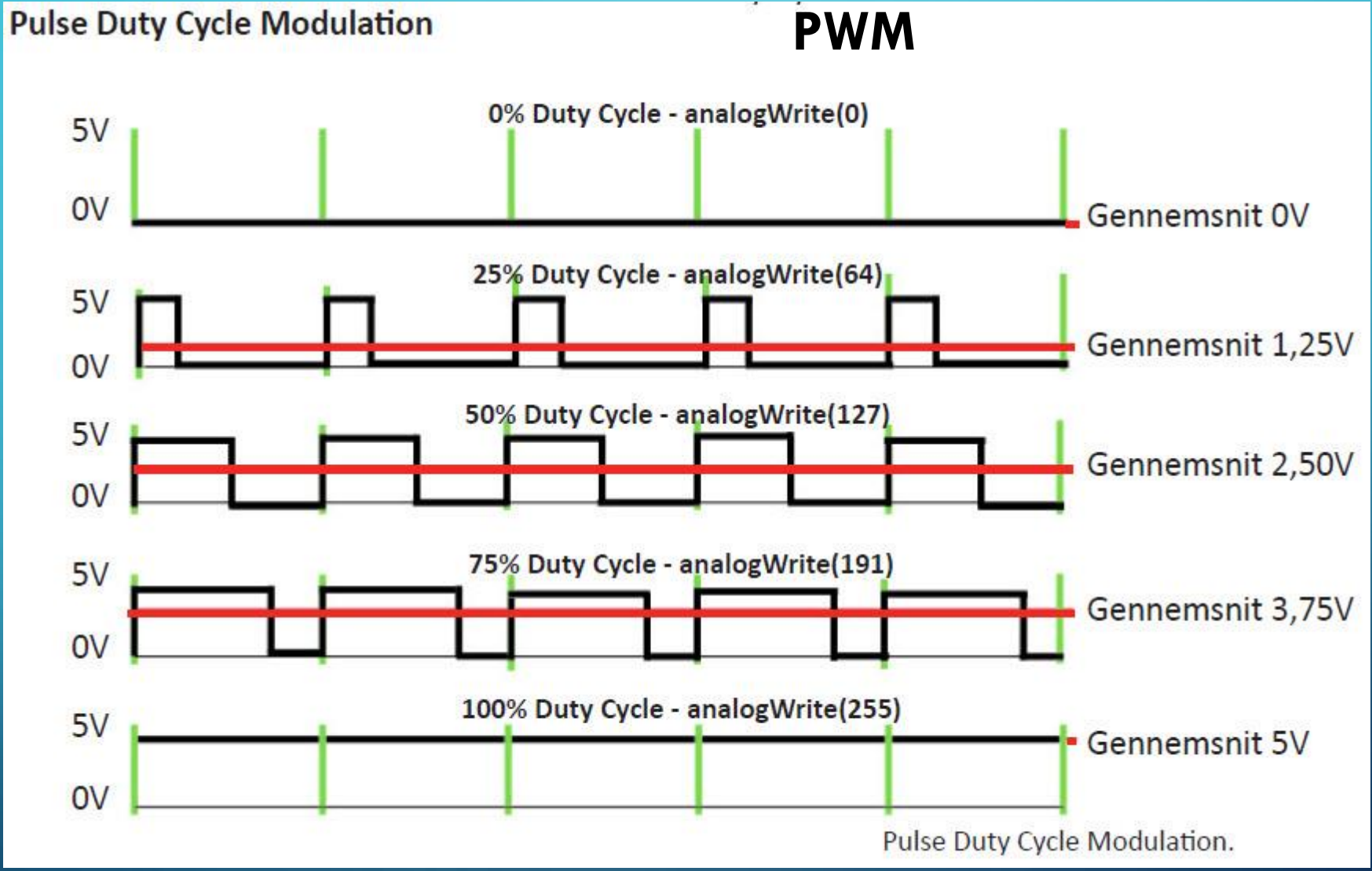


# Arduino UNO boardet

PWM pins 11-10-9-6-5-3



# Analog OUT?



`analogWrite (9, 0)`      pin 9 er altid OFF

`analogWrite (pin, value)`

`analogWrite (9, 255)`      pin 9 er altid ON



# Technical Specification



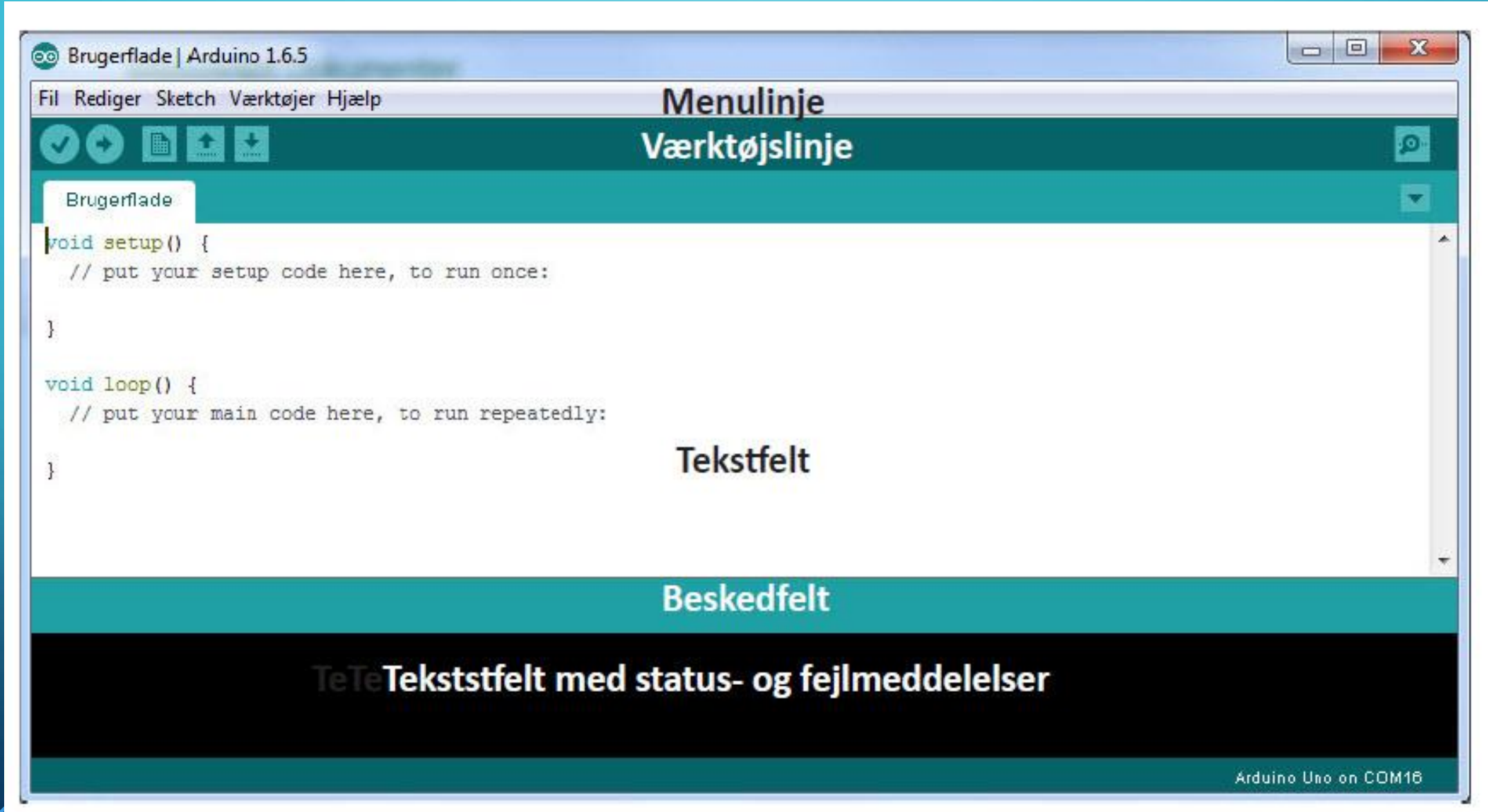
## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz





# ARDUINO IDE



# ARDUINO IDE

Kommentarer til programmet

```
/*  
*/  
//
```



Globale variable

Optrette variabler

```
Int led = 13
```



void setup()

```
{  
}
```

```
pinMode(led,OUTPUT)
```



void loop()

```
{  
}
```

```
digitalWrite(led, HIGH)
```

```
delay(1000)
```

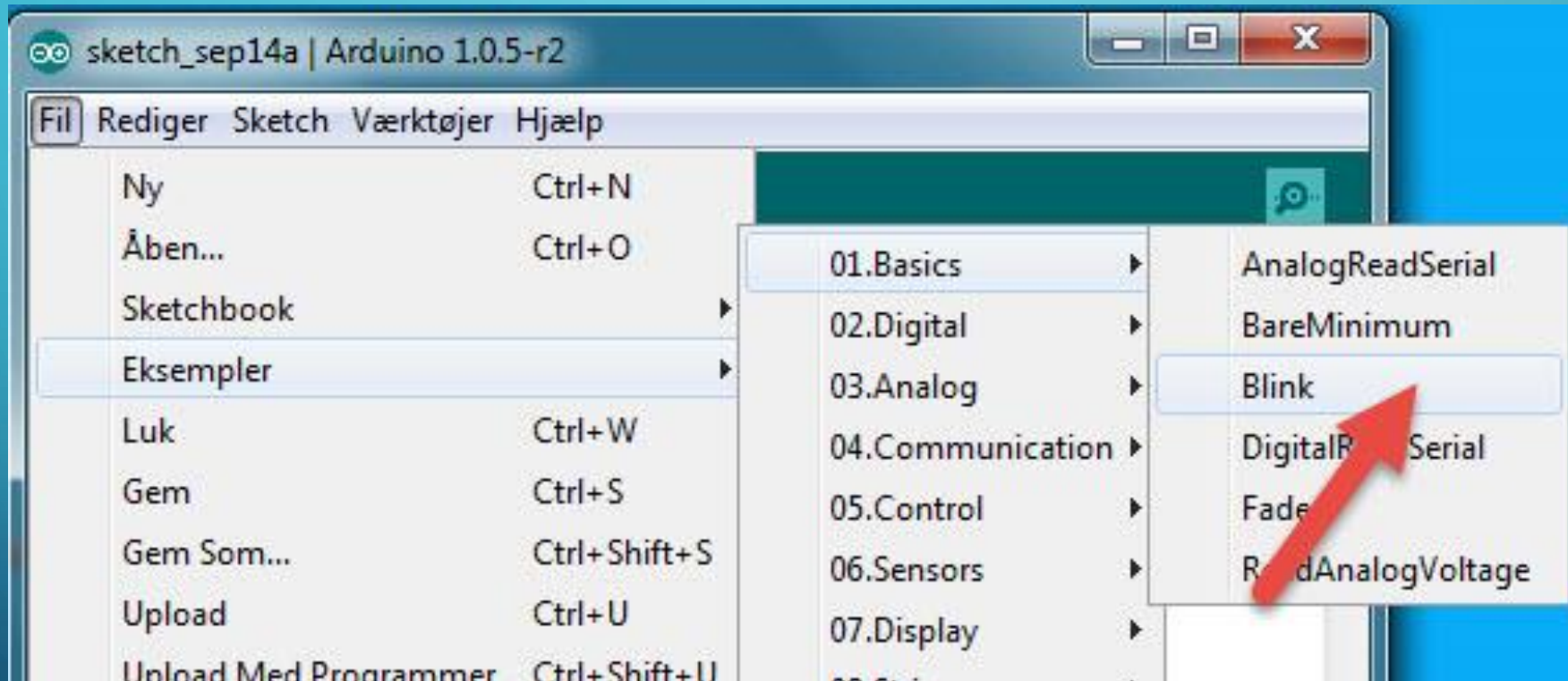
```
digitalWrite(led, LOW)
```

```
delay(1000)
```

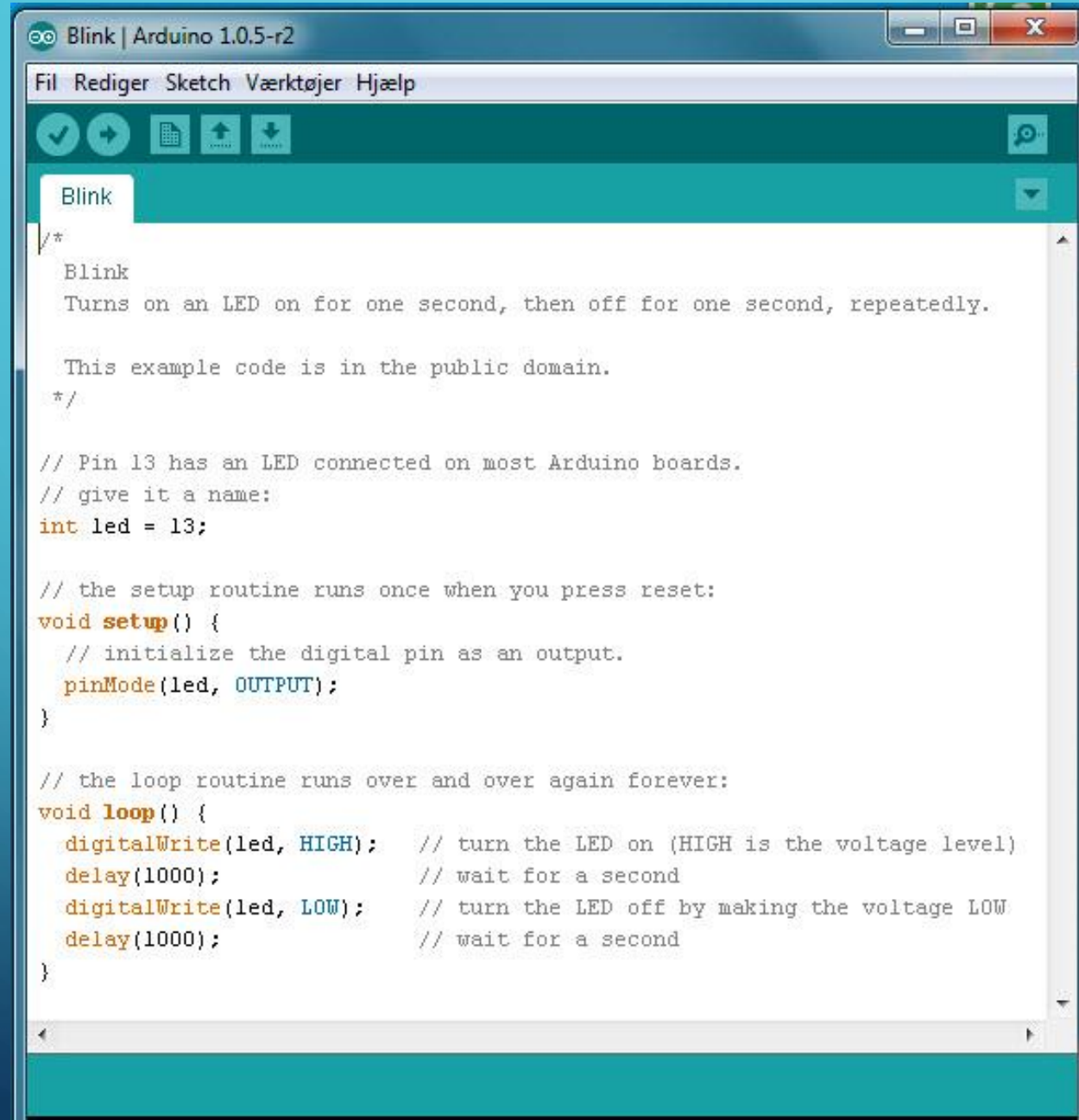




# "HELLO, WORLD!" – BLINK SKETCH



# "HELLO, WORLD!" – BLINK SKETCH

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.5-r2". The menu bar includes "Fil", "Rediger", "Sketch", "Værktøjer", and "Hjælp". The toolbar contains icons for a checkmark, a right arrow, a grid, an upload arrow, a download arrow, and a search icon. The sketch editor shows the following code:

```
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```



# ARDUINO IDE

`Int led=13`

`pinmode(n, INPUT)`

`pinmode(n, OUTPUT)`

`digitalWrite(n, HIGH)`

`digitalWrite(n, LOW)`

`digitalRead(n, HIGH)`

`digitalRead(n, LOW)`

`analogWrite(n, a)`

`analogRead(A0)`

`delay(x)`

`Serial.begin(9600)`

`Serial.println(analogRead(0))`

en variabel med et navn og en værdi

sæt pin n som input

sæt pin n som output

sæt pin til 5V

sæt pin til 0V

læser 5V på pin n

læser 0V på pin n

skriver værdien a til pin n. a ligger i intervallet 0-255

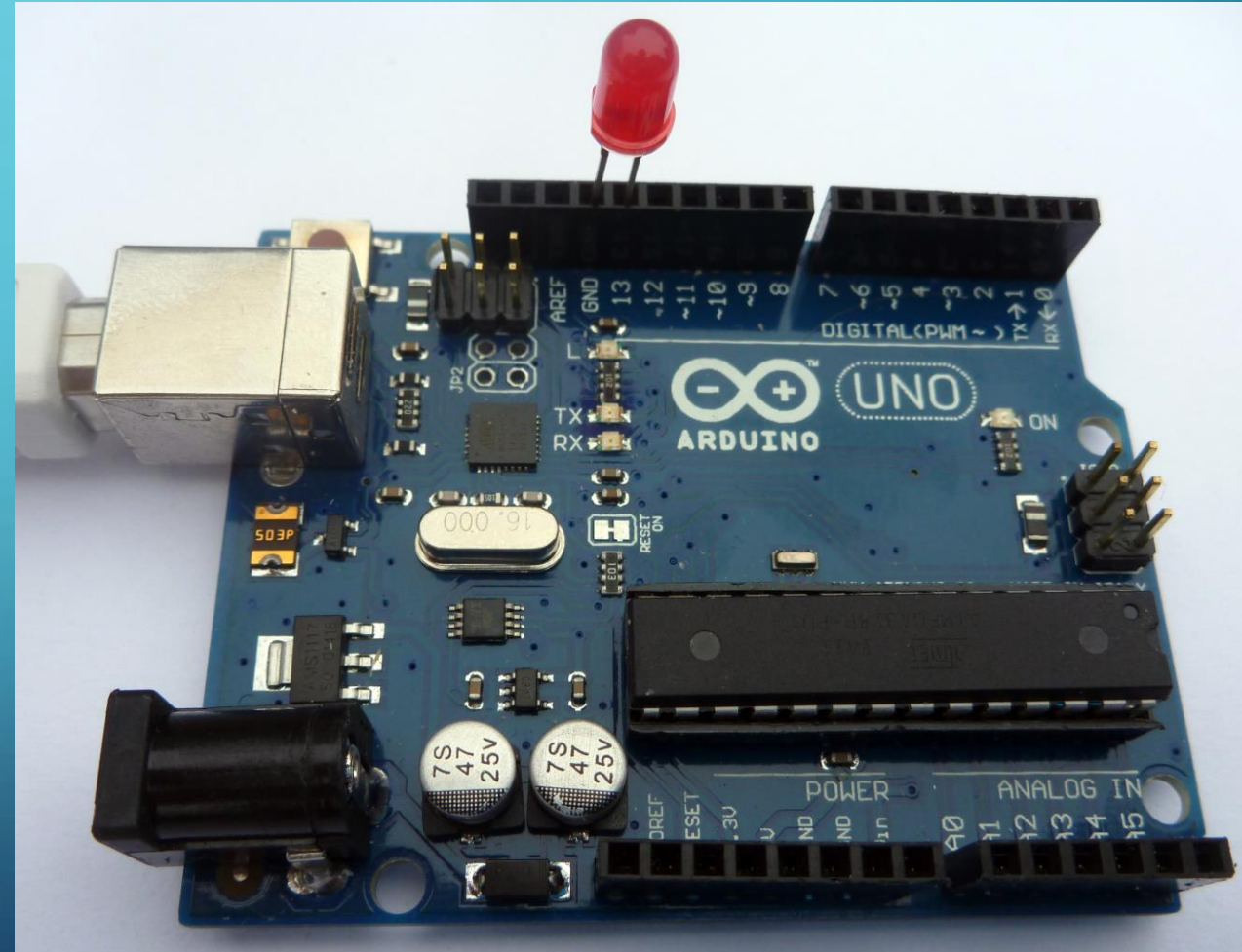
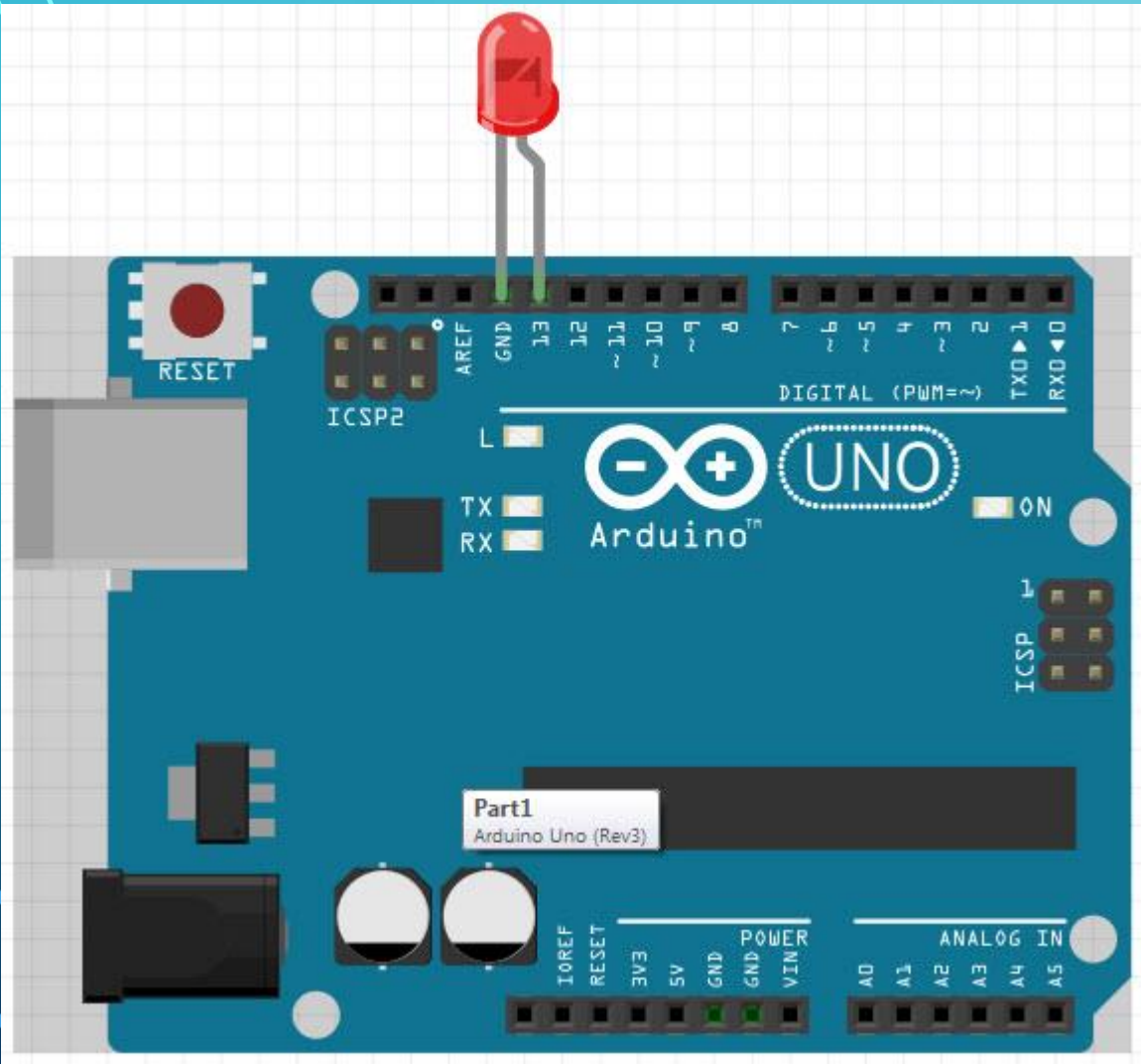
læser værdien af den analoge pin A0 i intervallet 0-1023

sæt programmet på pause i x millisekunder x = 0 til 65.535

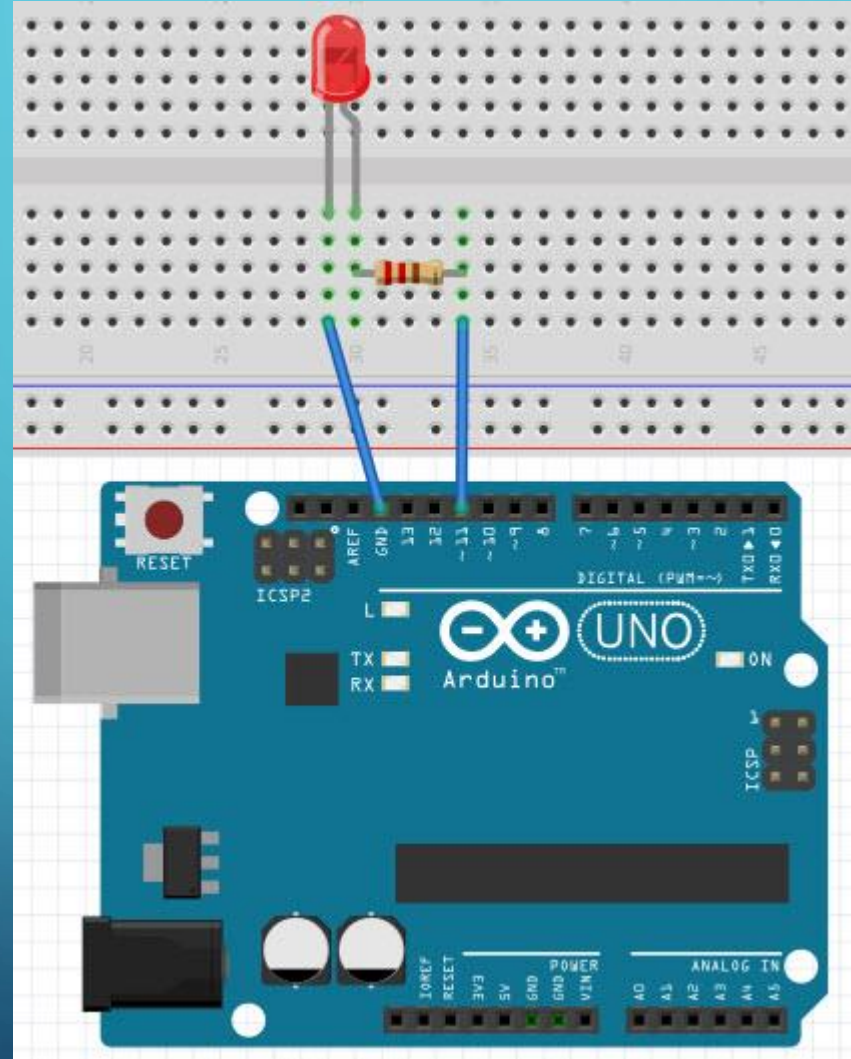
åbner seriel port med en datahastighed på 9600 bps

Skriver værdien fra A0 på skærmen

”HELLO, WORLD!”



# "HELLO, WORLD!"

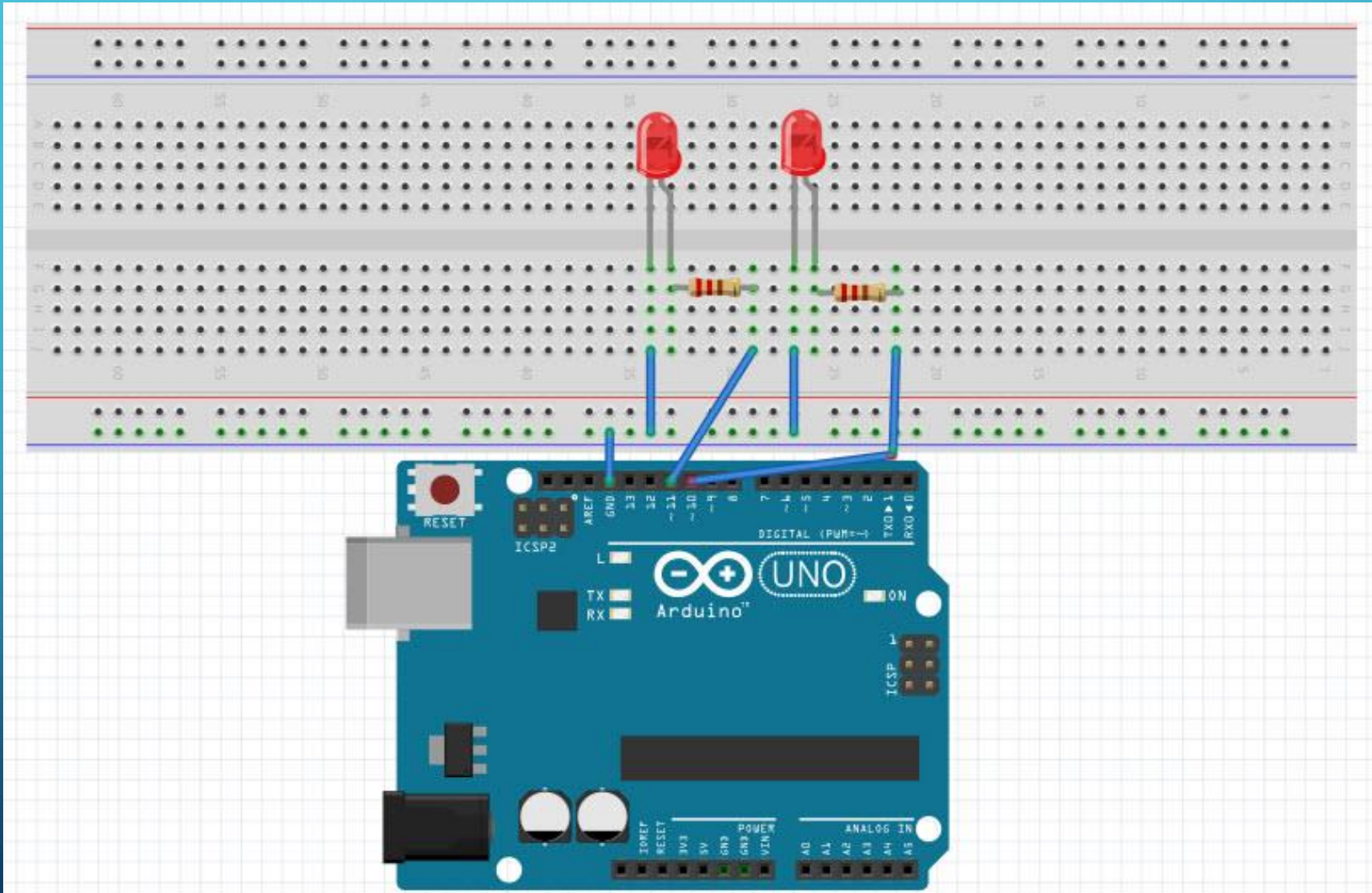




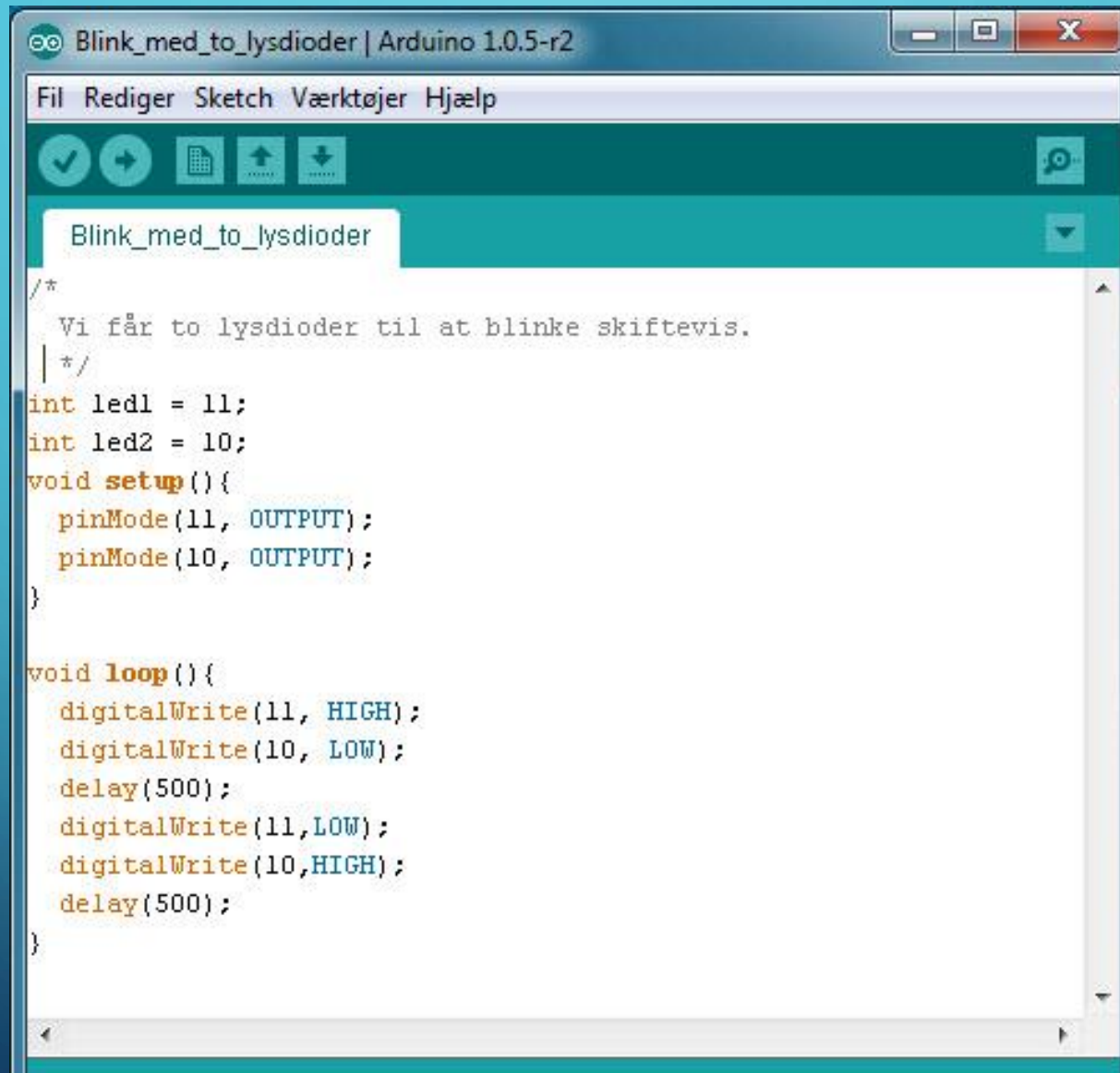
# BLINK MED TO LYSDIODER - TORONTOANLÆG



# BLINK MED TO LYSDIODER



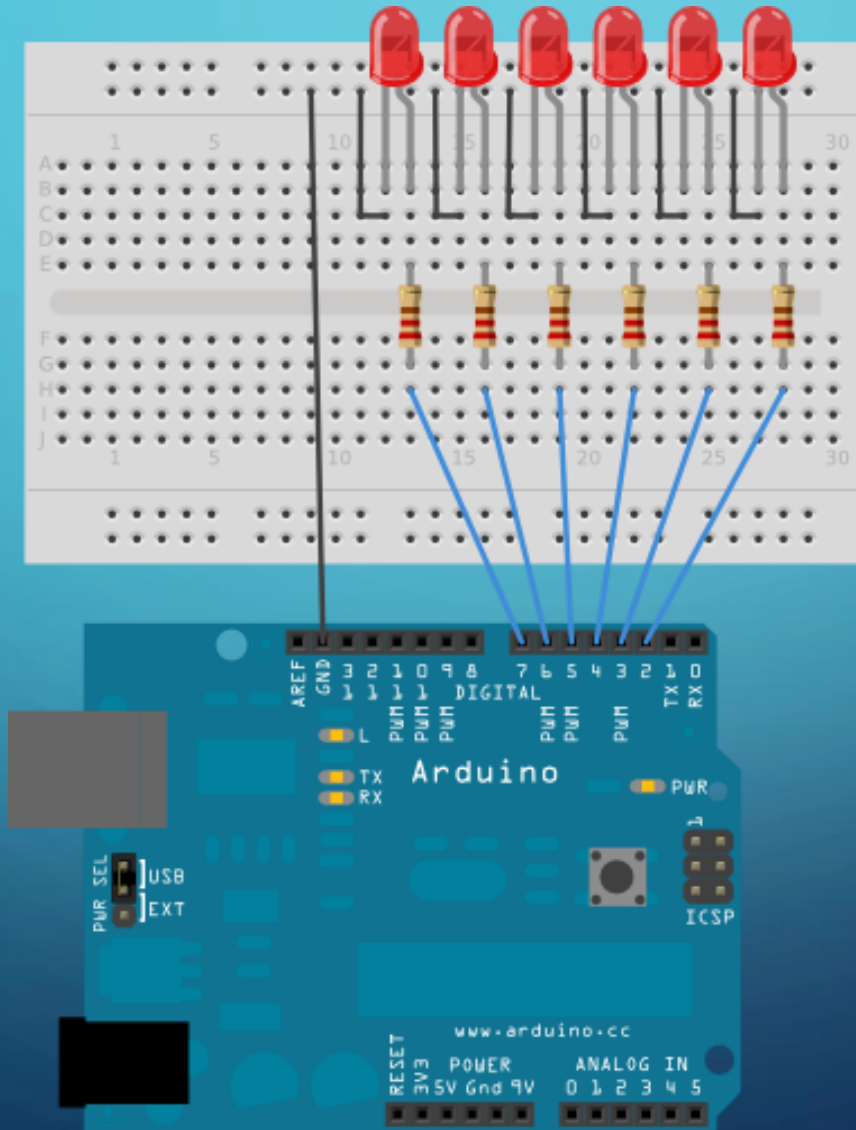
# BLINK MED TO LYSDIODER - SKETCH

A screenshot of the Arduino IDE interface. The window title is "Blink\_med\_to\_lysdioder | Arduino 1.0.5-r2". The menu bar includes "Fil", "Rediger", "Sketch", "Værktøjer", and "Hjælp". The toolbar contains icons for saving, running, uploading, and downloading. The sketch name is "Blink\_med\_to\_lysdioder". The code in the editor is as follows:

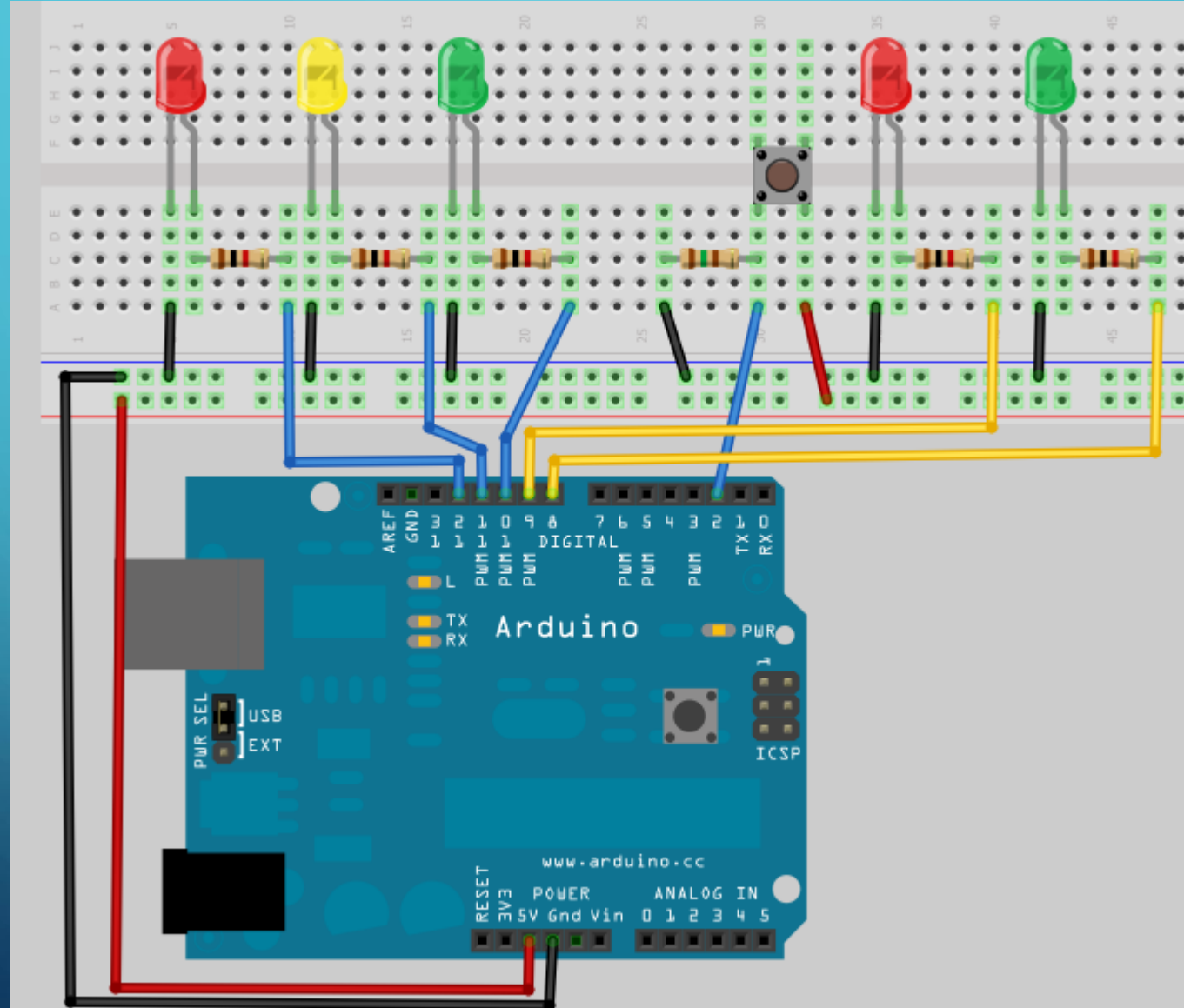
```
/*  
 Vi får to lysdioder til at blinke skiftevis.  
 */  
int led1 = 11;  
int led2 = 10;  
void setup(){  
  pinMode(11, OUTPUT);  
  pinMode(10, OUTPUT);  
}  
  
void loop(){  
  digitalWrite(11, HIGH);  
  digitalWrite(10, LOW);  
  delay(500);  
  digitalWrite(11, LOW);  
  digitalWrite(10, HIGH);  
  delay(500);  
}
```



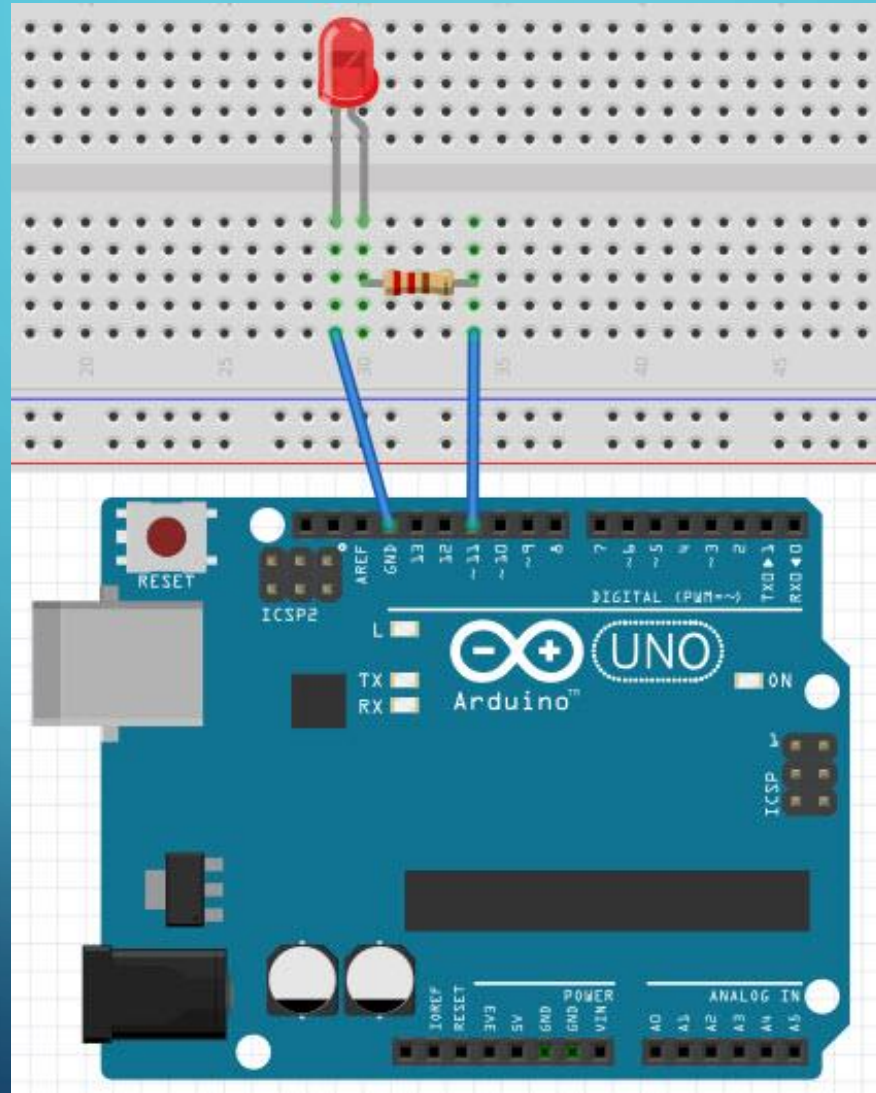
# BLINK MED 6 LYSDIODER - HARDWARE



# LYSREGULERINGER

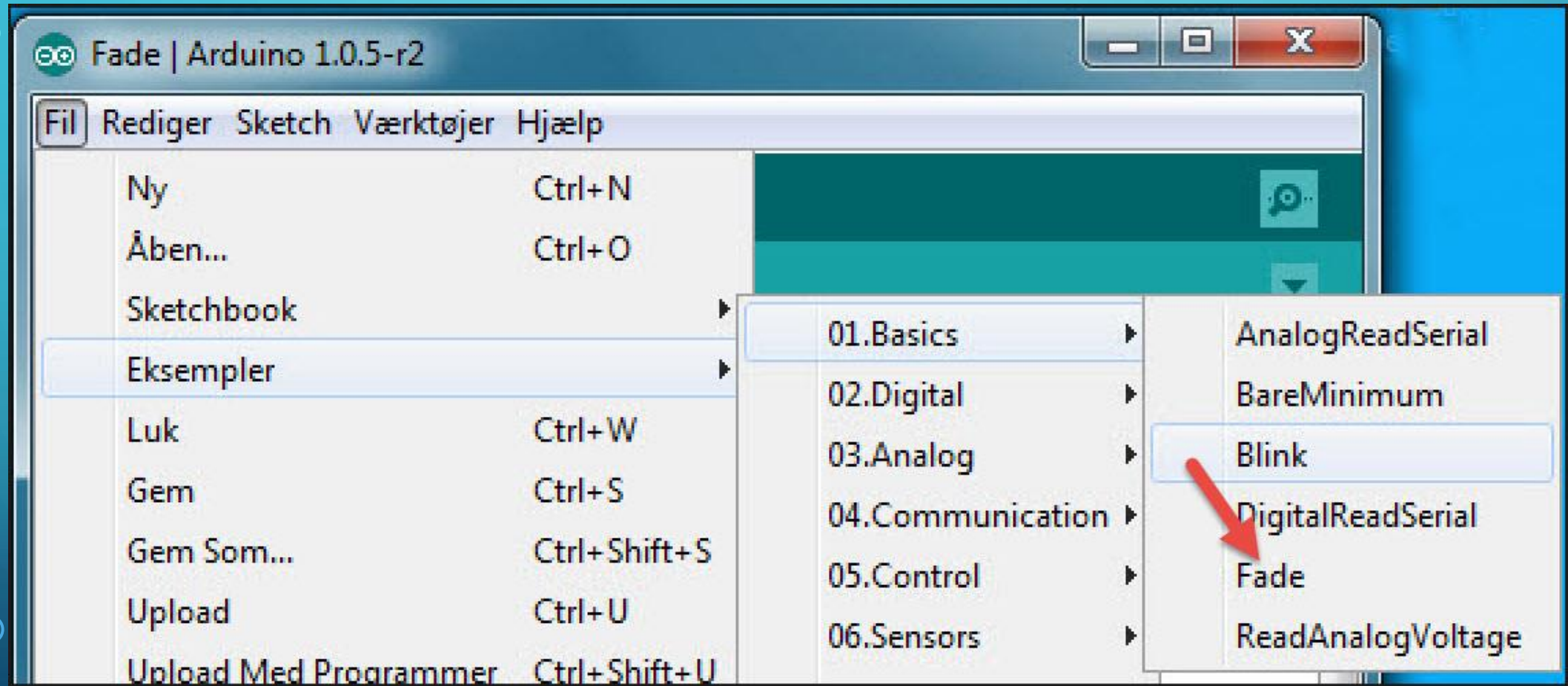


# FADE EN LYSDIODE





# FADE EN LYSDIODE



# FADE EN LYSDIODE

```
Fade | Arduino 1.0.5-r2
Fil Rediger Sketch Værktøjer Hjælp

Fade $

int led = 11;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 11 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 11:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

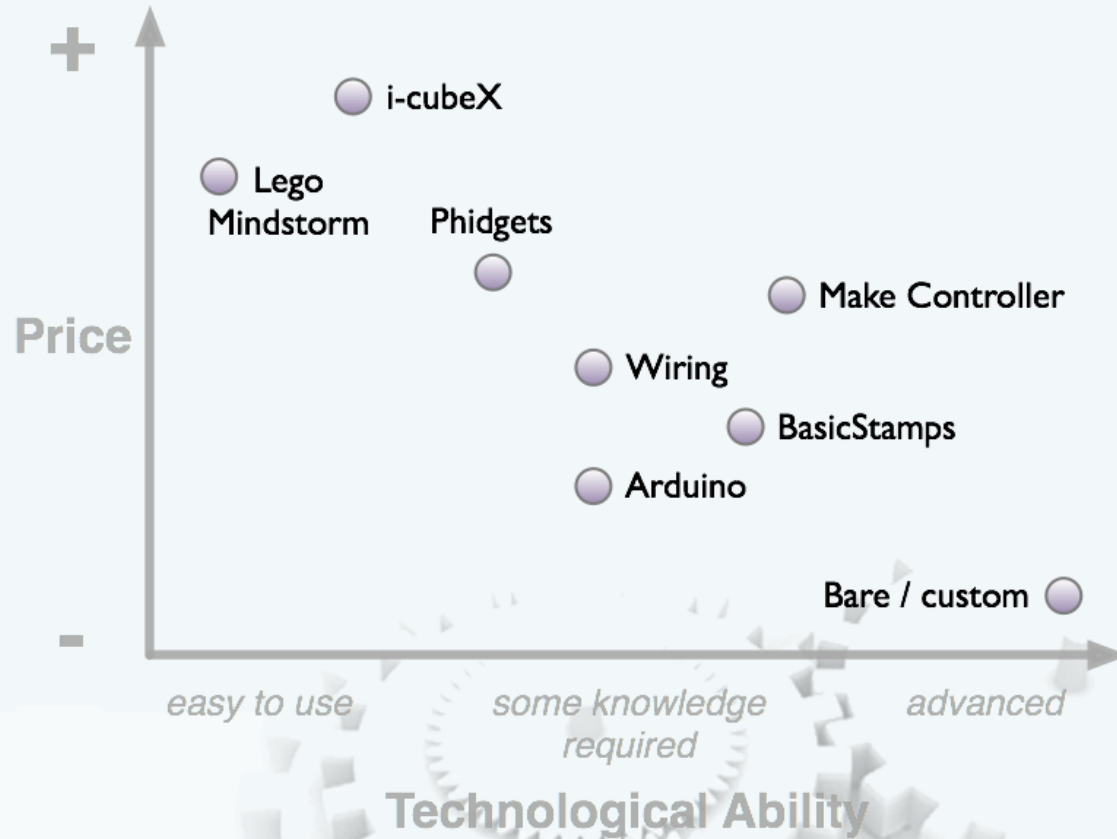
  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

# EKSISTERENDE HARDWARE

Existing Toolkits



irish open source  
technology conference





# SENSORER - AKTUATORER

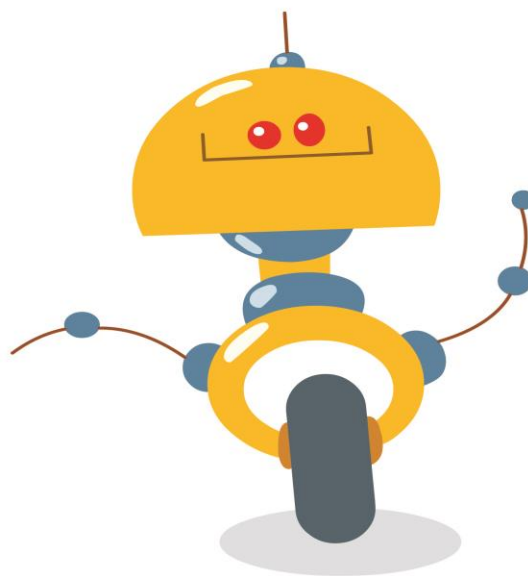
## Arduino

Sensorer - Input



Output - Aktuatorer

Lyssensor  
Afstandssensor  
Kamera  
PIR-sensor  
Lydsensor(mikrofon)  
Temperatursensor  
Fugtighedssensor  
Kollisionssensor  
GPS  
Tryksensor  
Kraftsensor  
Alkohol - sensor  
CO-sensor  
m. fl

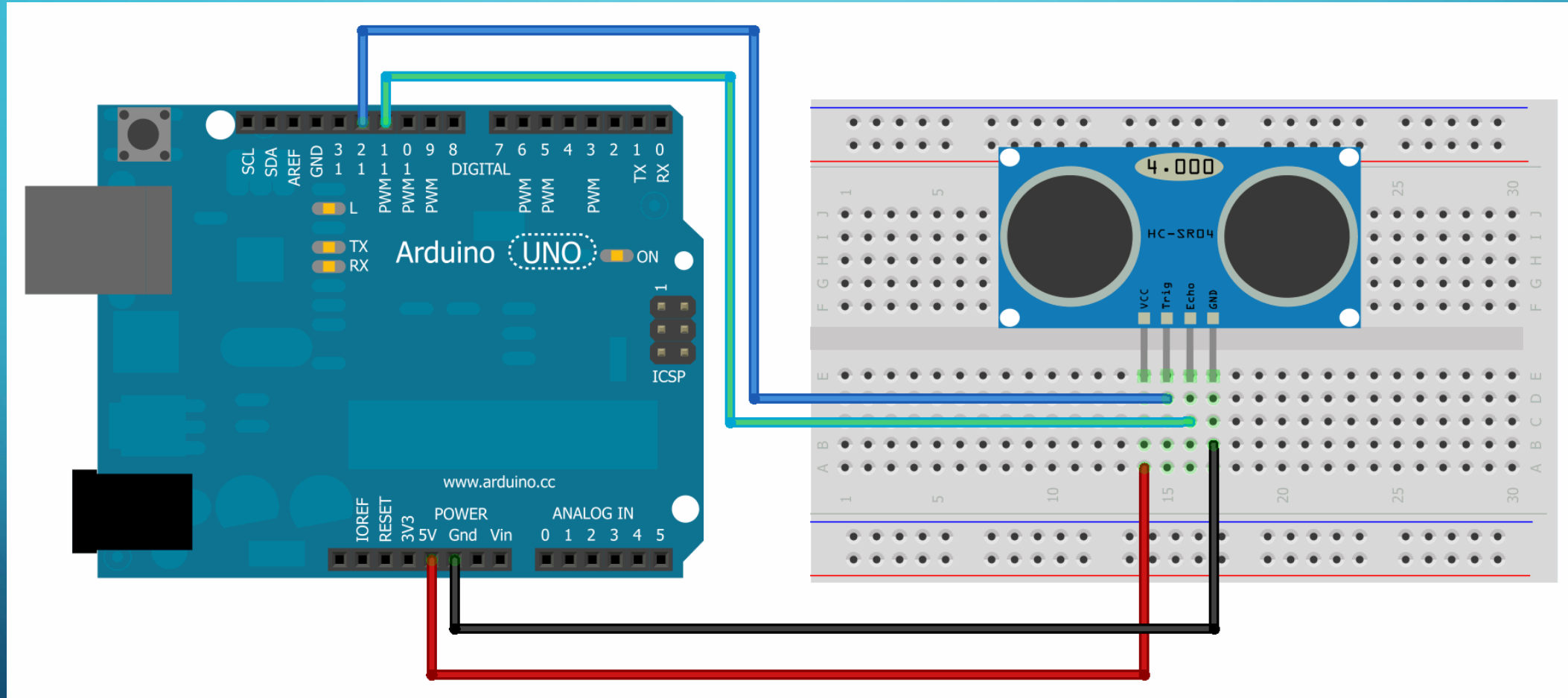


Motorer  
Larvefødde  
Hjul  
Gribearme (hænder)  
Display (skærm)  
Lyd  
Lys  
m. fl

# AFSTANDSSENSOR



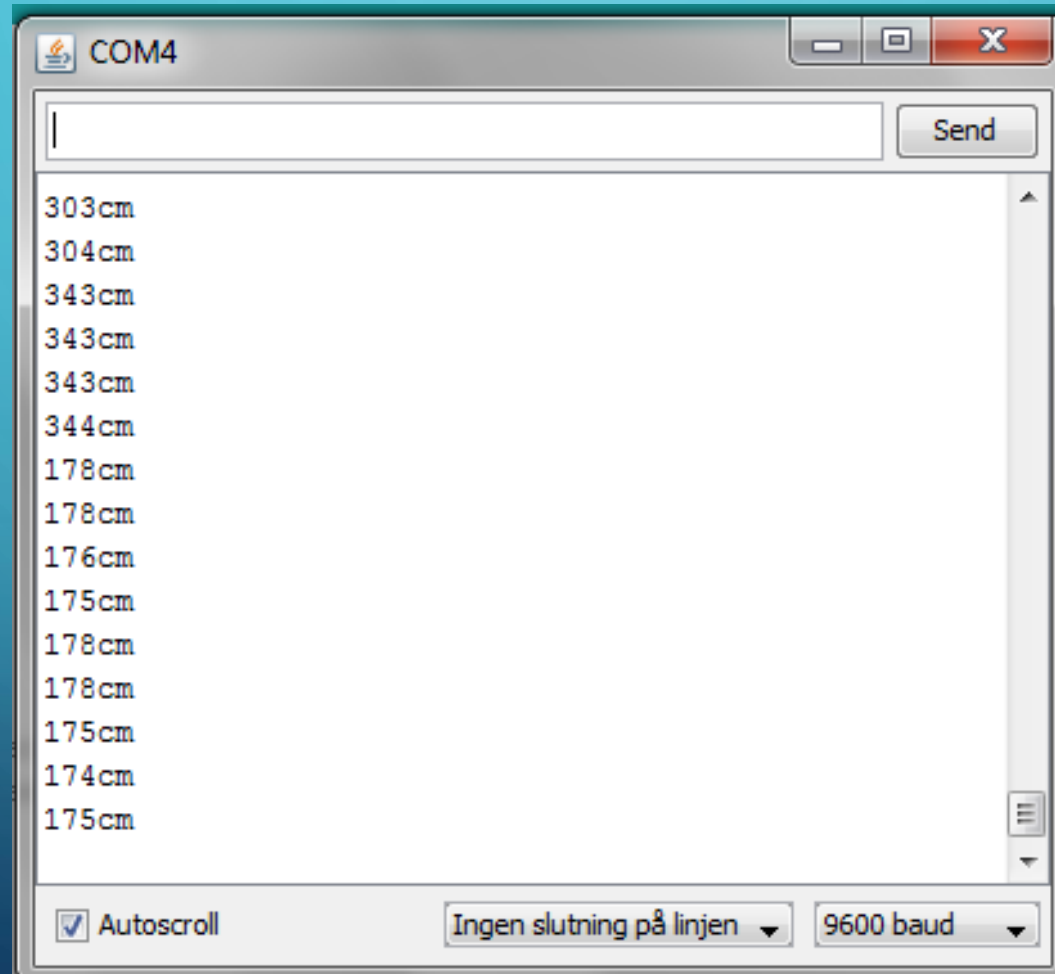
# AFSTANDSSENSOR - HARDWARE



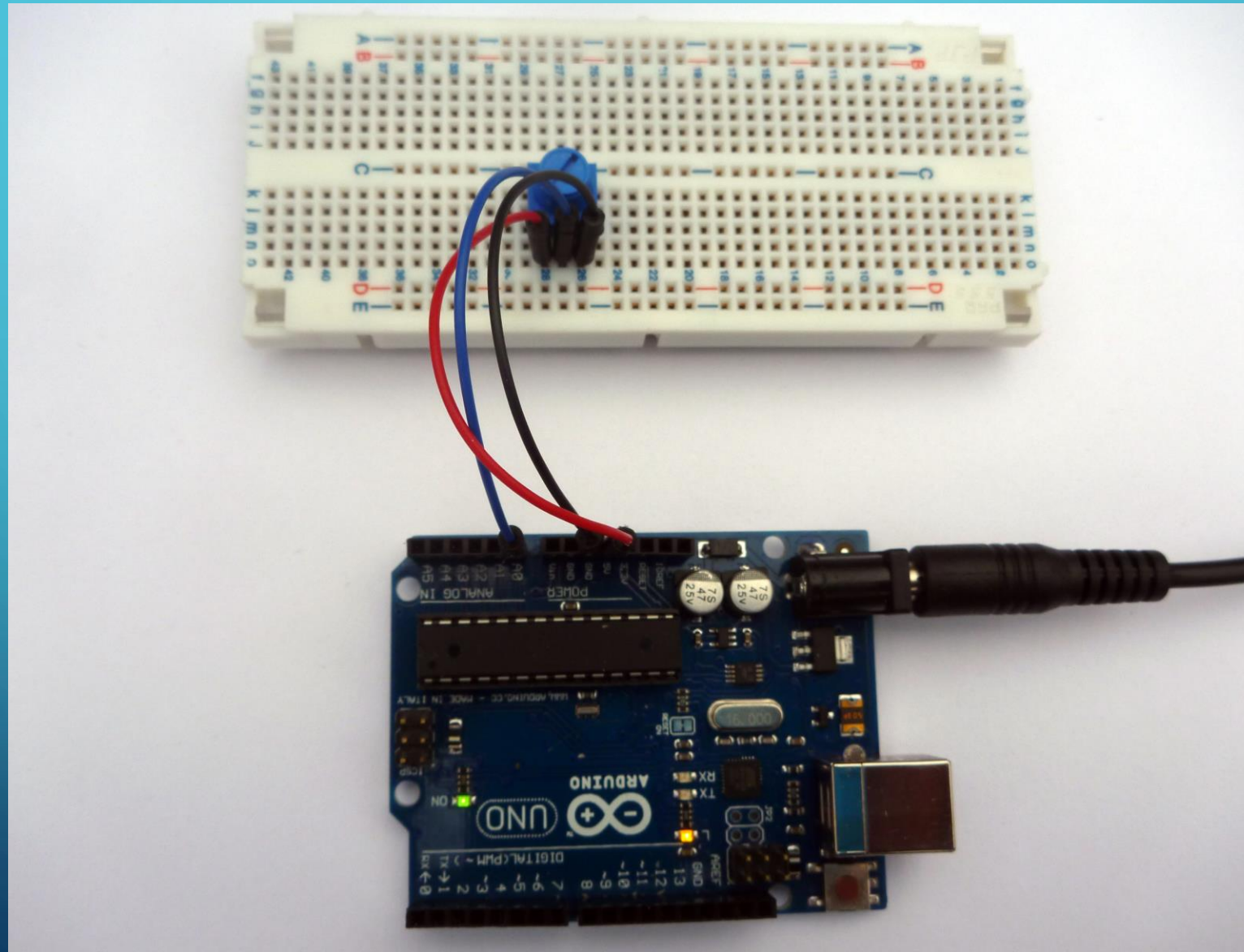


# AFSTANDSENSOR

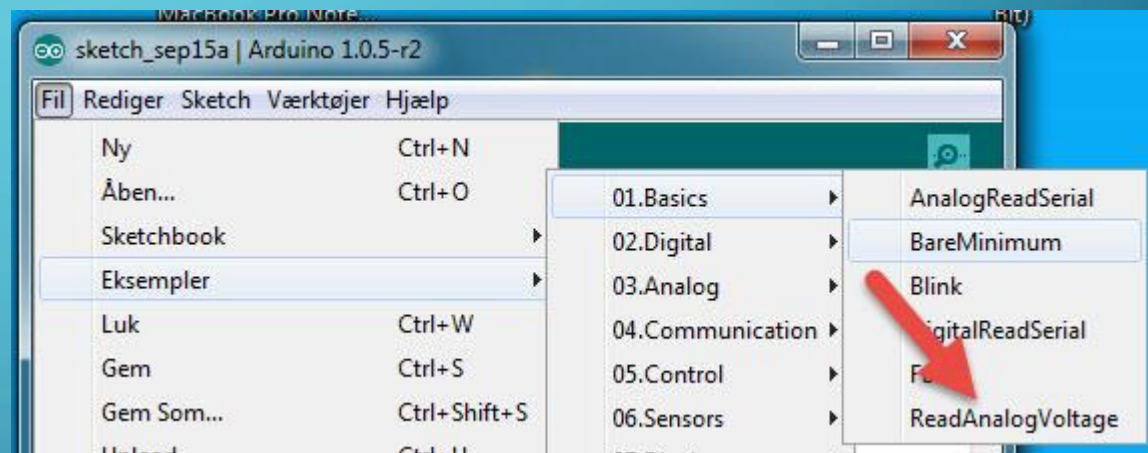
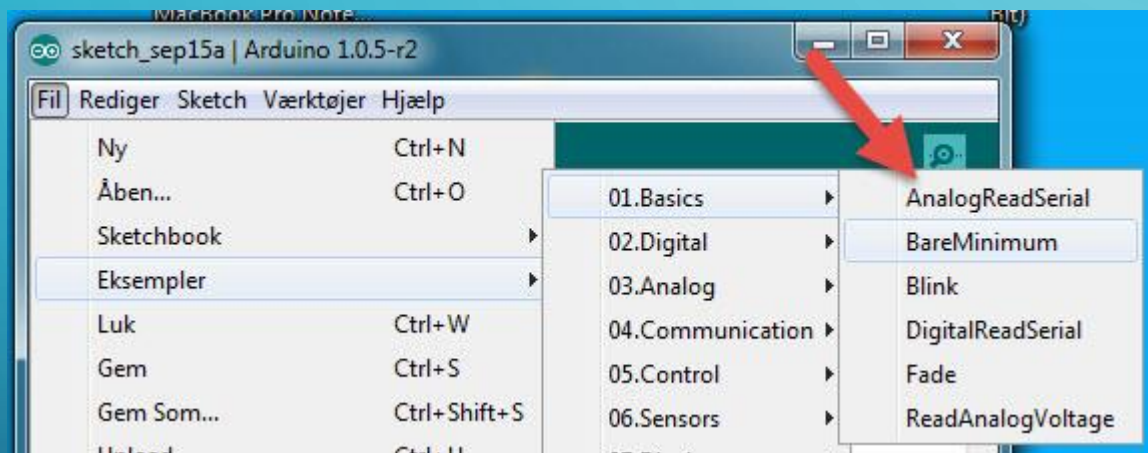
## Data fra Serial Monitor



# POTENTIOMETER



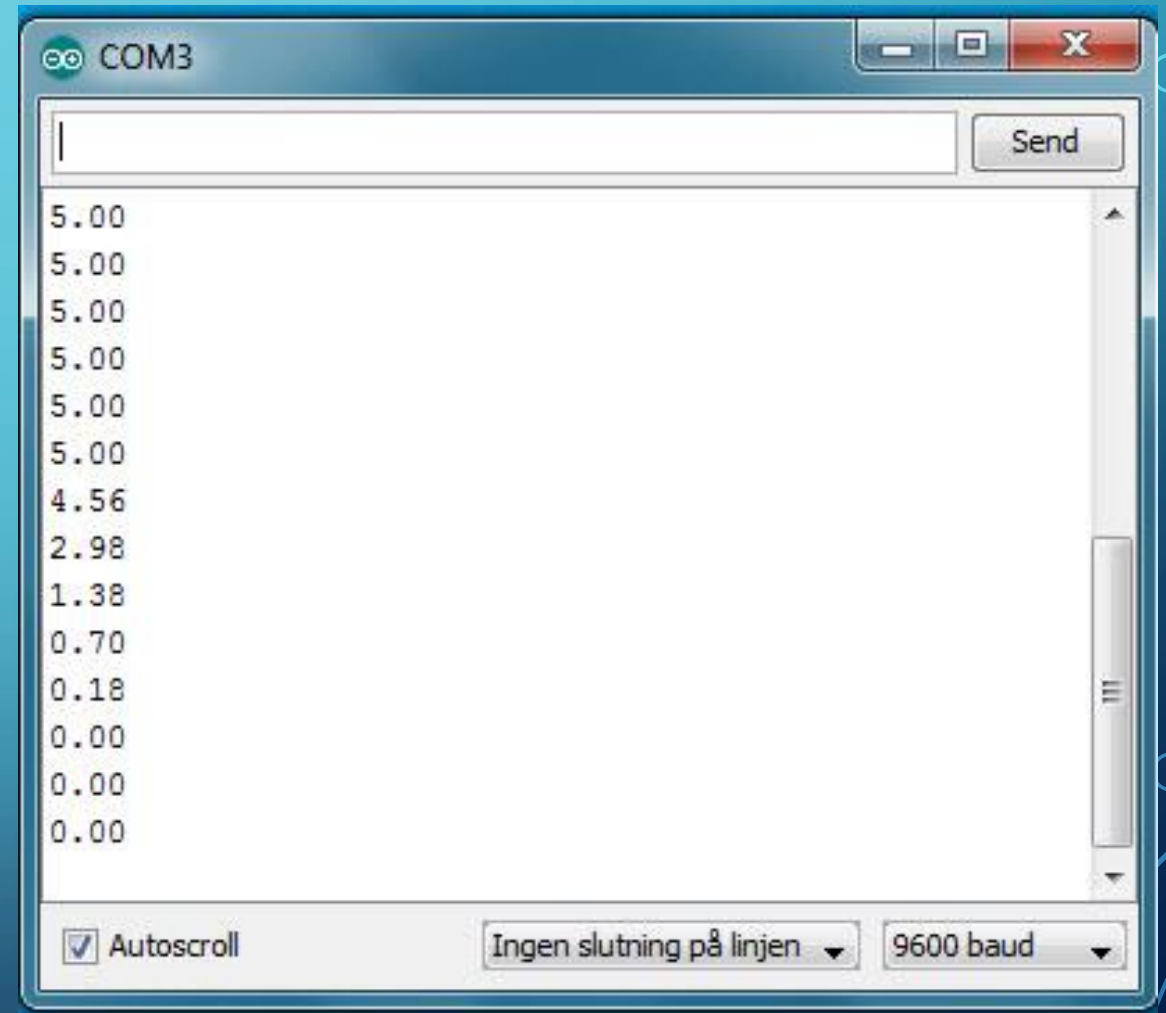
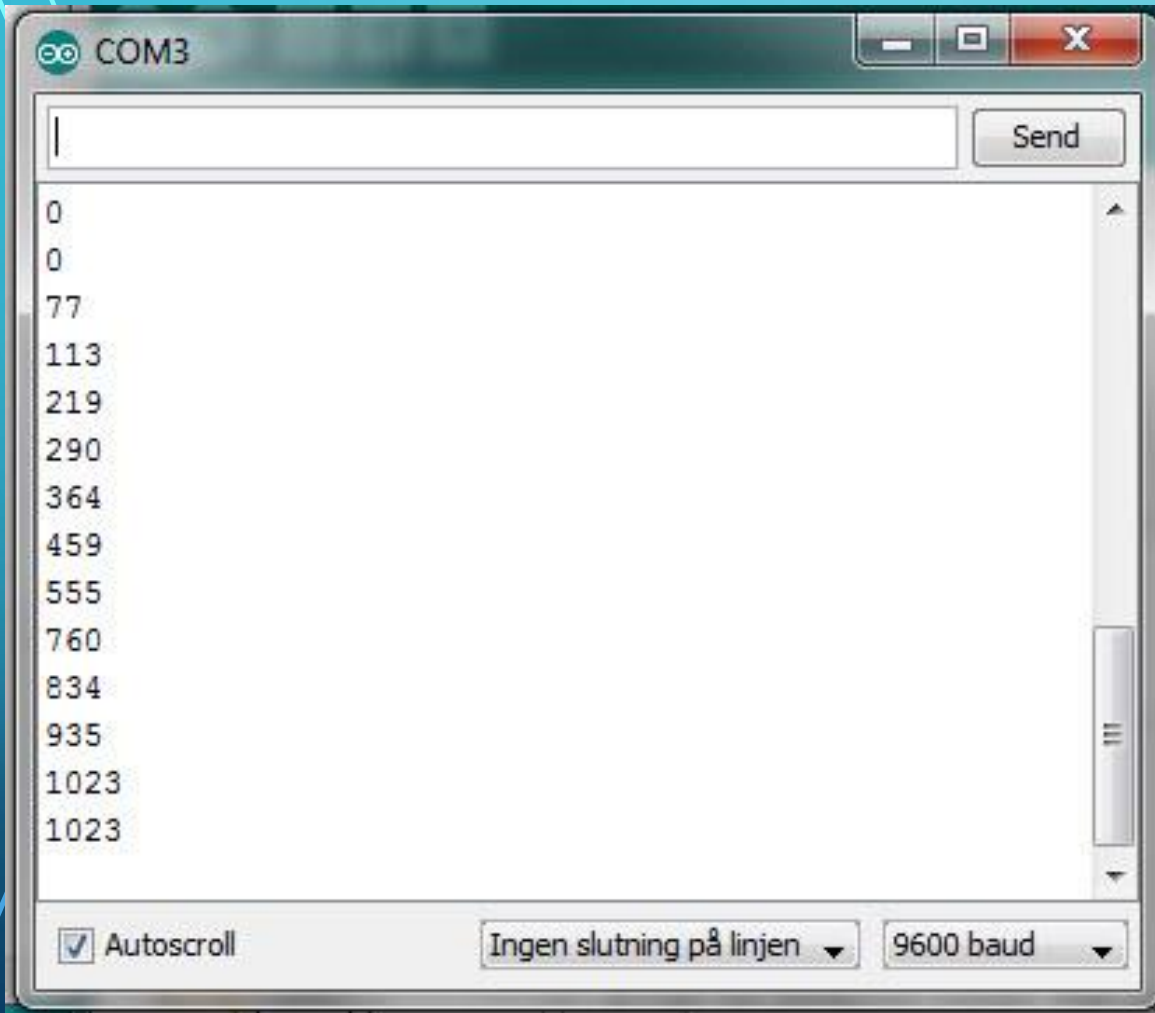
# POTENTIOMETER - SKETCHES





# POTENTIOMETER

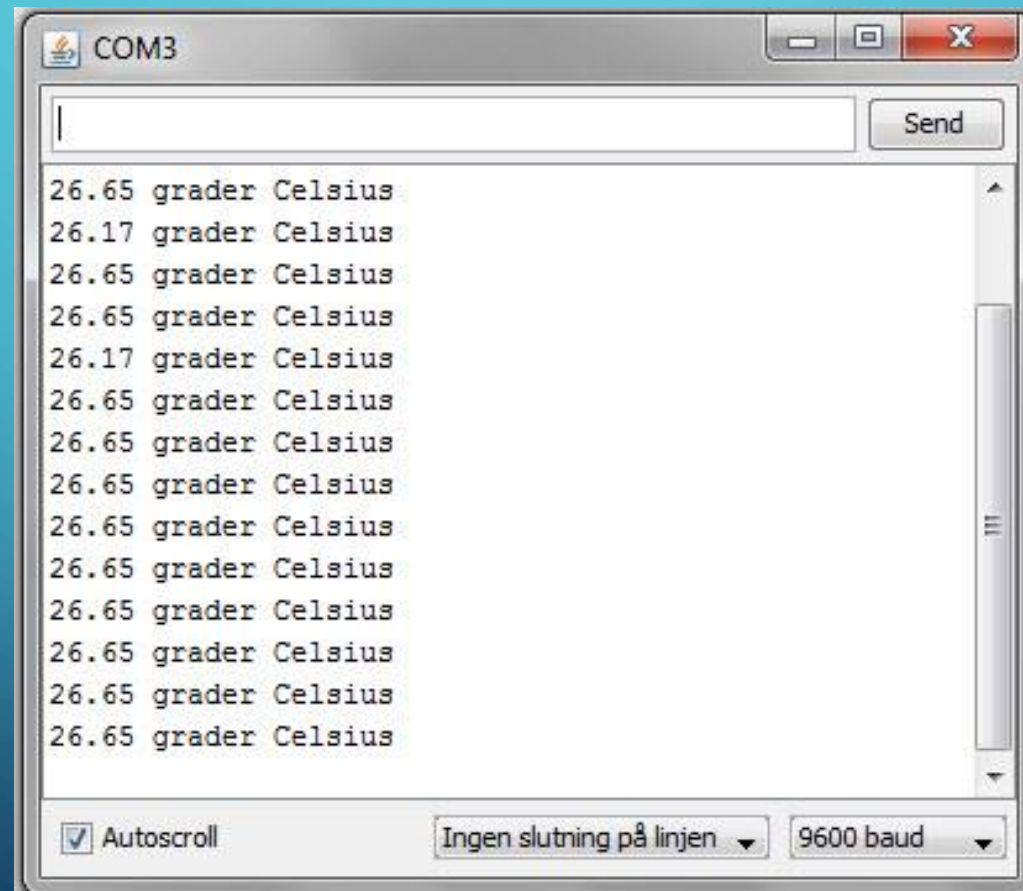
## Data fra Serial Monitor





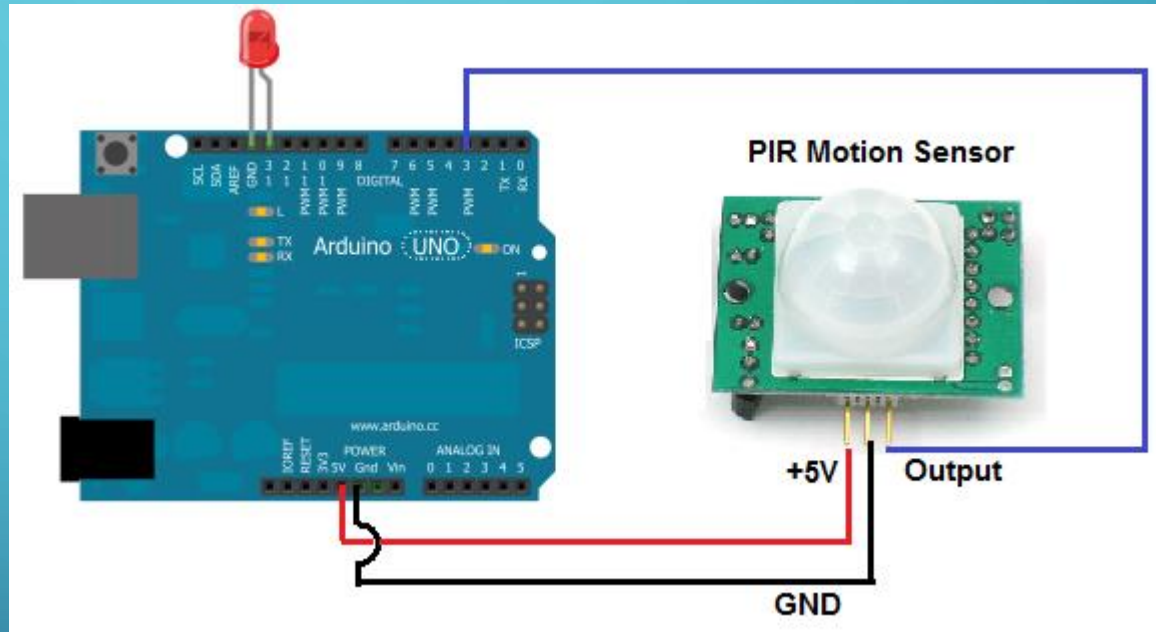
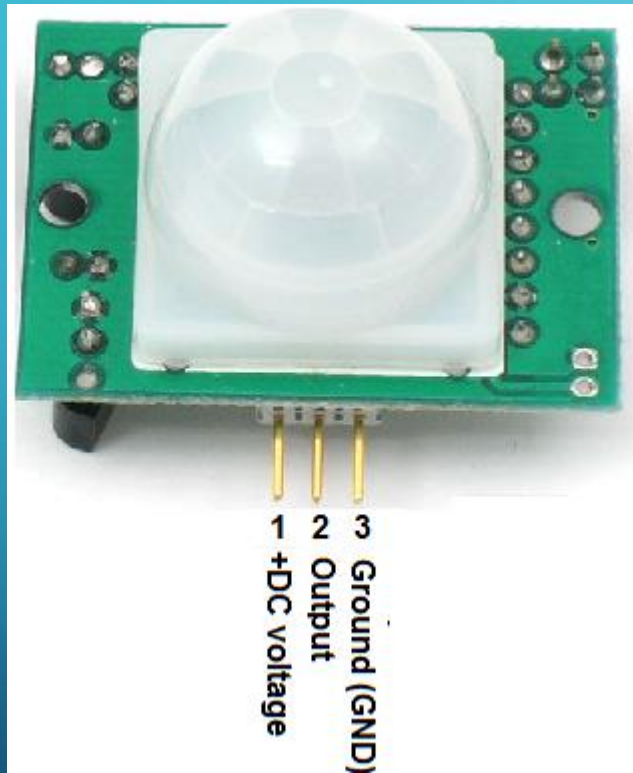
# TEMPERTURSENSOR

## Data fra Serial Monitor

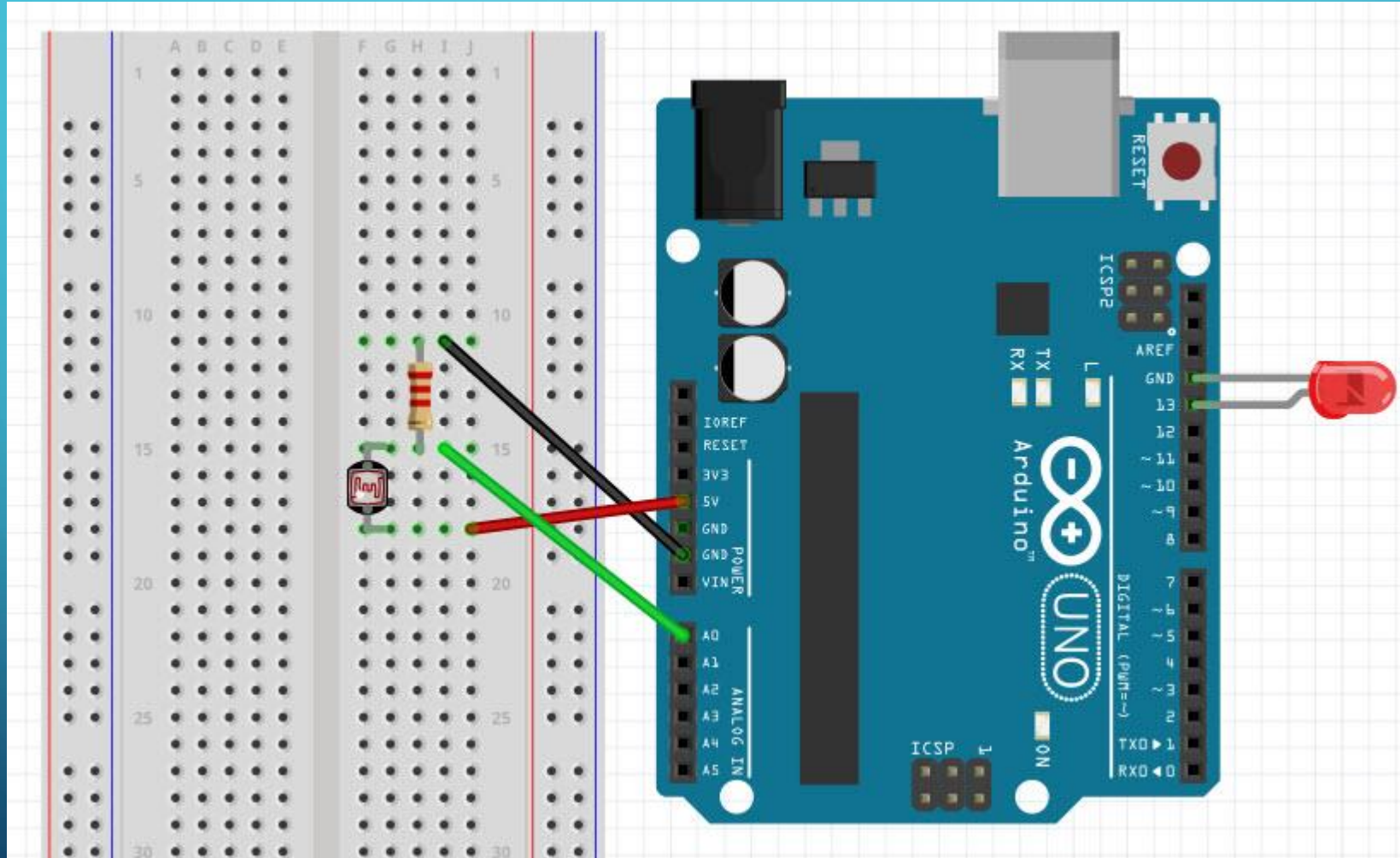




# BEVÆGELSESENSOR

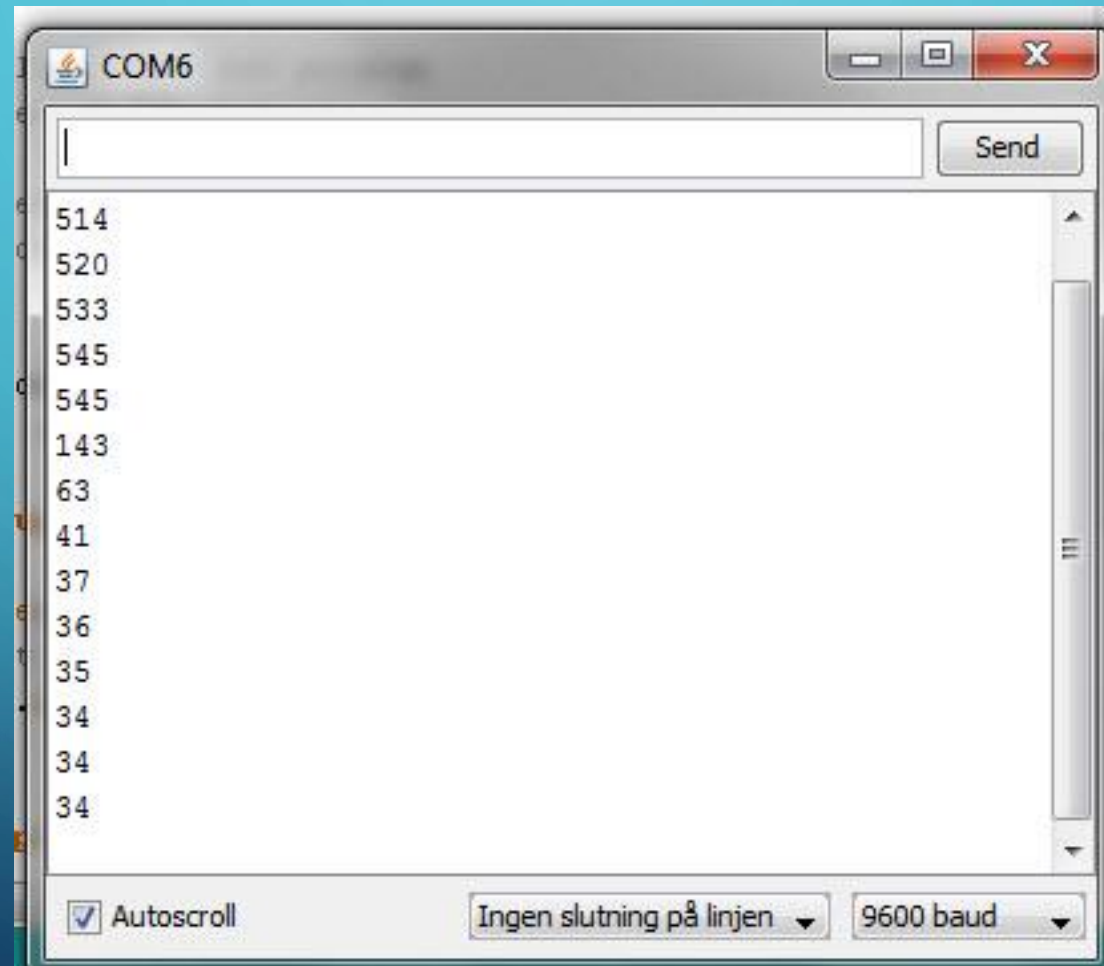


# PHOTORESISTOR - LDR



# PHOTORESISTOR

## Data fra Serial Monitor





The background is a dark blue gradient. In the corners, there are decorative white line-art elements resembling circuit traces or a network diagram, with small circles at the end of the lines.

Spørgsmål? – hvis ikke

Hands on