

A blue Arduino Uno microcontroller board is shown from a top-down perspective, angled slightly. A silver USB cable is connected to the board's USB port. The board features a central ATmega328P microchip, various surface-mount components, and a breadboard-style pin header. A black rectangular overlay contains the title text.

ARDUINO CRASHCOURSE

– ZERO TO HERO

HVEM ER JEG?



- **Jacob Bechmann Pedersen**

- Diplomingeniør Elektronik (AU, 2019)
- Startede med Arduino i 2014
- Frivillig i Coding Pirates 2016-2018
- Underviser på MakerCamp siden 2018
 - 12-16 y/o "Inventors"
- Afholdt kurser i Arduino for IDA, StudyNow, Herning Kommune, etc.
- Embedded electronics engineer på DTU
 - Robotter og autonome systemer

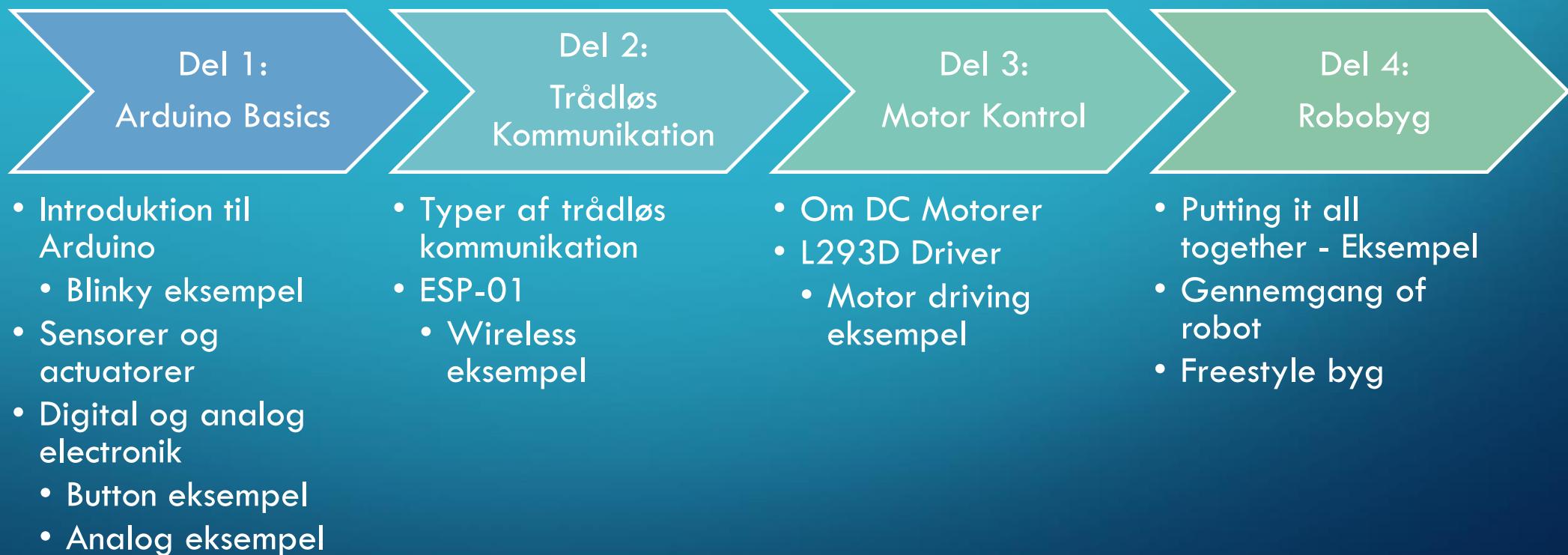
AGENDA

- 16:00-17:00 : Arduino Basics – Blinky
- 17:00-18:00 : Trådløs kommunikation med Arduino
- 18:00-18:45 : Aftensmad - Pizza og sodavand
- 18:45-19:45 : Motor control og robotbyg
- 19:45-20:00 : Afrunding og tak for denne gang!

FORMÅL MED DENNE WORKSHOP

- At give en **HURTIG** introduktion til de essentielle dele af Arduino:
 - Noget hands-on succes
 - Møder med nogle af udfordringerne
- Prøve mulighederne med platformen
- Inspiration til hvordan I selv kan bruge Arduino

DAGENS INDHOLD



FØR VI STARTER

- Koden og eksemplerne for denne workshop er at finde på:
 - <https://github.com/iakop/ArduinoCrashcourse>
 - Det er en god idé at holde denne åben



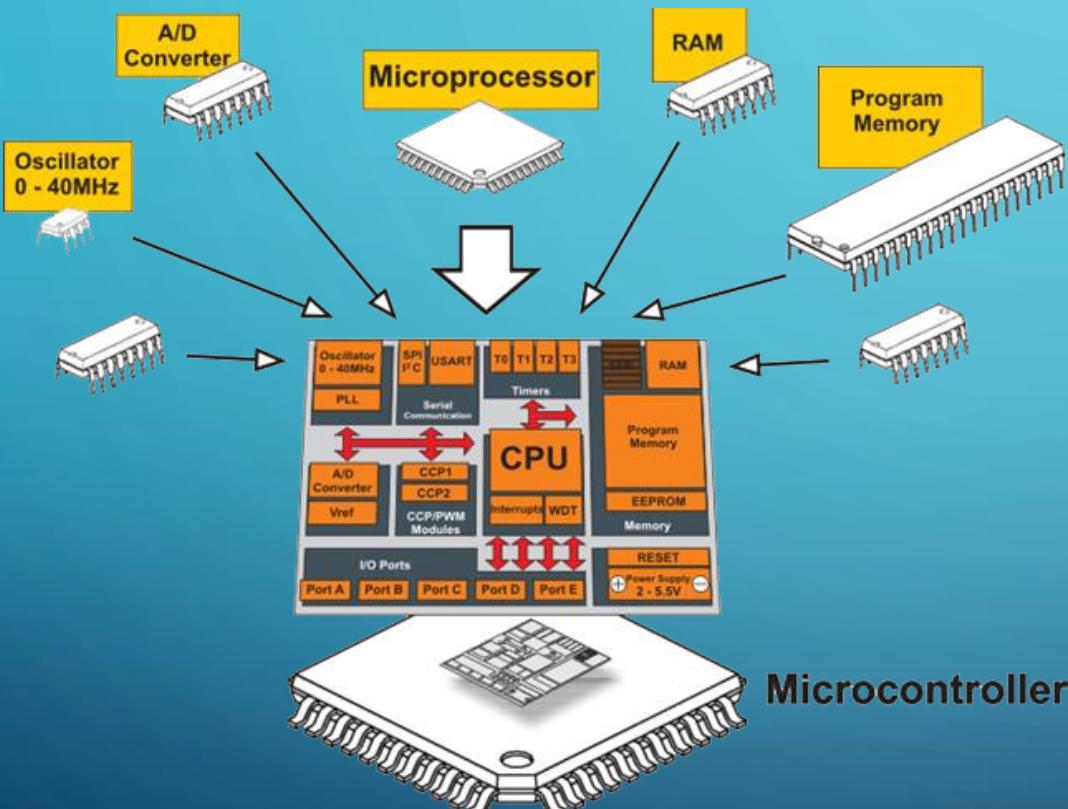
The background image shows an Arduino Uno microcontroller board. A black rectangular overlay box is centered on the board. Inside this box, the text "DEL 1:" is on top and "ARDUINO BASICS" is below it, both in large white capital letters. The entire assembly is set against a solid teal background. Two light blue circuit line diagrams are overlaid on the image: one on the left side pointing towards the top center, and another on the right side pointing towards the bottom center.

DEL 1:

ARDUINO BASICS

INTRO TIL ARDUINO

HVAD ER EN MICROCONTROLLER?

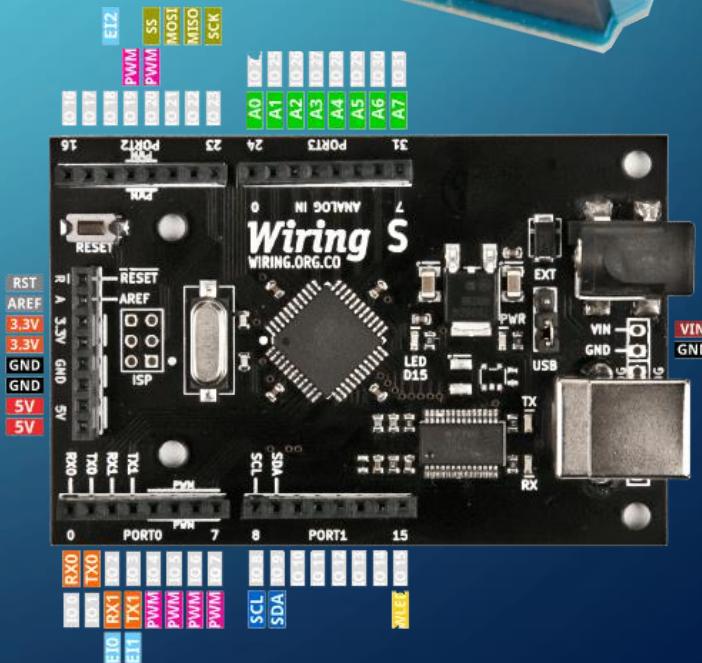


- Forkortet MCU (MicroController Unit)
- Skaber hverdagens magi
- Binder software og hardware sammen
- Ikke bare processorer:
 - Indeholder et helt system
 - Storage, memory, program memory, ADC mm.
- Lette at integrere i elektroniske kredsløb

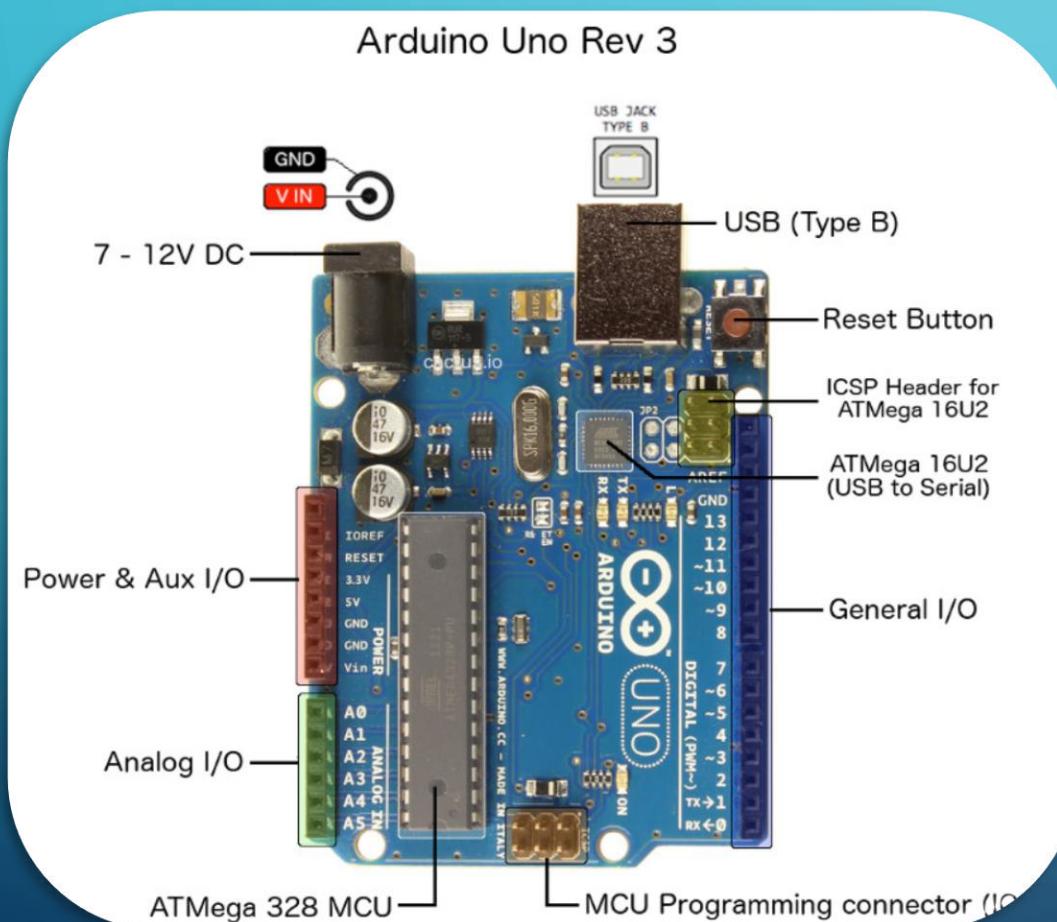
Figure from: Introduction to the World of MicroControllers – Mikroelektronika
URL: <https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c/introduction-to-the-world-of-microcontrollers>

ARDUINO PLATFORMEN

- En fork of Wiring af Hernando Barragán
 - Master's Thesis i Interaction Design
 - Baseret på Processing (programmeringssprog)
 - Gør microcontroller udvikling lettere
- Mest brugte platform til MCU prototyping
- Simpel, intuitiv programmering
- Mange eksempler på kode tilgængelige
 - Mange software libraries



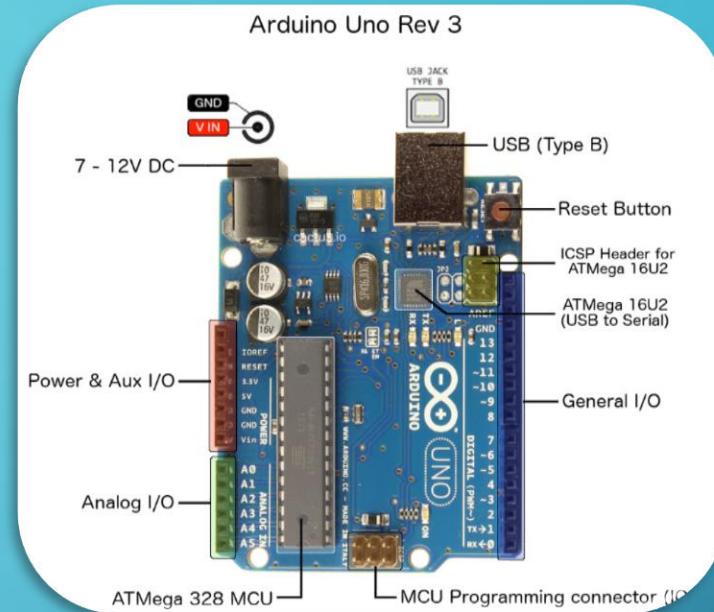
ARDUINO PLATFORMEN



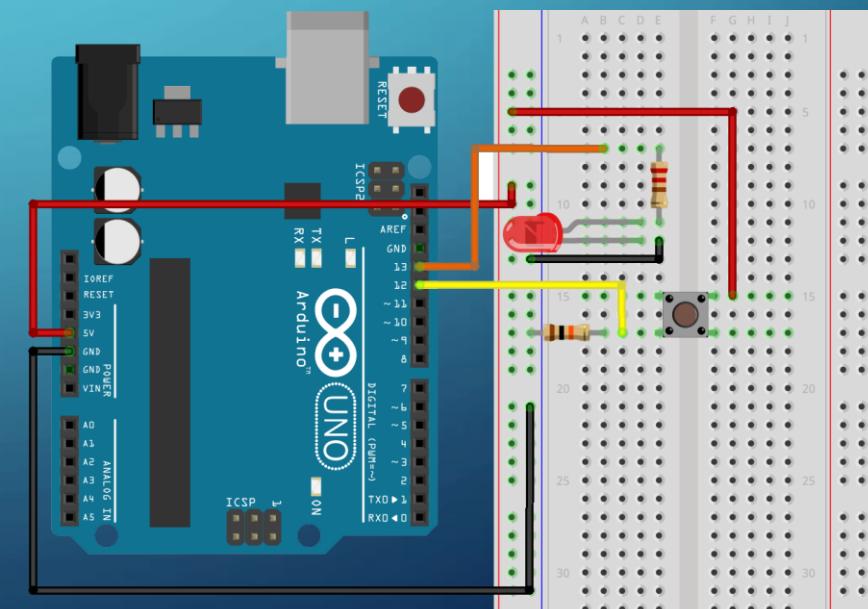
- Microcontrolleren (Atmega328) er hjernen bag
- DC og USB B Port til strøm og programmering
- Pins:
 - RED Strømforsyning, reset osv.
 - GREEN Analog input/output
 - BLUE General Purpose input/output
- Og andet godt

ARDUINO PLATFORMEN

- Vi forbinder den til elektronikken gennem dens pins
- Pin signaler styres af vores programmer (turn on/off, etc.)
- Forbindes oftest gennem et "breadboard"



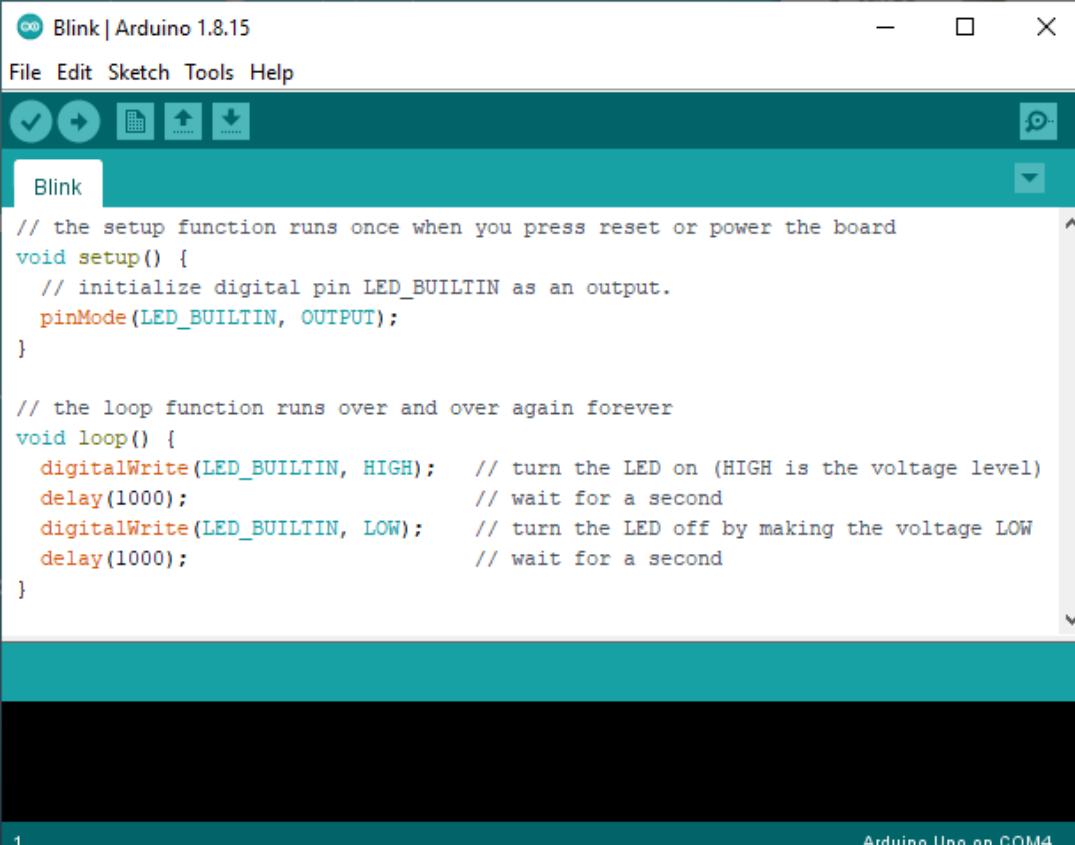
Arduino and its pins



Arduino connected to electronics on a solderfree breadboard

ARDUINO PLATFORMEN

- Programmers gennem dets "integrated development environment" (IDE)
- Programmerne indeholder altid:
 - `setup()` – commands to run only once
 - `loop()` – commands that will loop forever
- Skrevet i C/C++ programmeringssproget



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.15". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations. The main workspace displays the "Blink" sketch code. The code consists of two functions: `setup()` and `loop()`. The `setup()` function initializes the digital pin LED_BUILTIN as an output. The `loop()` function alternates the LED state between HIGH and LOW every second, using `delay(1000)` to wait for a second between each state change.

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

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

The official Blink program example in Arduino IDE

HVORFOR ARDUINO?

- Observer dette program:
- Barebones AVR program
- Man skal kende til processoren, dens port og pin adresser
- Er svært at læse og forstå
- Men det sparar da noget plads!

```
1 //Physical LED pin
2 #define LEDWRITEPORT PORTB
3 #define LEDPIN 5
4 //Physical button pin
5 #define BTNREADPORT PIND
6 #define BTNPIN 3
7
8 int main() {
9
10    //Initialize pins
11    DDRB |= (1 << LEDPIN); //ledPin set to output
12    DDRD &= ~(1 << BTNPIN); //btnPin set to input
13
14    while(1){
15        //If button is pressed LED on
16        if((PIND & (1 << BTNPIN)) >> BTNPIN == 1){
17            PORTB |= (1 << LEDPIN);
18        }
19        //Else, LED off
20        else{
21            PORTB &= ~(1 << LEDPIN);
22        }
23    }
24 }
```

Sketch uses 148 bytes (0%) of program storage space. Maximum is 32256 bytes.

Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.

HVORFOR ARDUINO?

- Og tag så dette program:
 - Til at læse for mennesker
 - Simple operationer
 - Tydelig struktur
- Kræver mere plads på processoren
 - Fair tradeoff for lettere programmering

```
1 const int ledPin = 13; //Physical LED pin
2 const int btnPin = 3; //Physical button pin
3
4 void setup() {
5     //Intialize pins
6     pinMode(ledPin, OUTPUT); //ledPin set to output
7     pinMode(btnPin, INPUT); //btnPin set to input
8 }
9
10 void loop() {
11     //If button is pressed LED on
12     if(digitalRead(btnPin) == HIGH){
13         digitalWrite(ledPin, HIGH);
14     }
15     //Else, LED off
16     else{
17         digitalWrite(ledPin, LOW);
18     }
19 }
```

KOM I GANG:

- Download og installér Arduino IDE
- Skaf versionen til jeres platform
 - For jeres egen skyld og resten af verden – download IKKE Windows Store versionen!
- Eksemplerne fra denne workshop ligger på Github:
<https://github.com/iakop/ArduinoCrashcourse>



The screenshot shows the Arduino IDE download page. At the top left is the Arduino logo (an infinity symbol with a plus sign). To its right is the text "Arduino IDE 1.8.16". Below the logo is a brief description: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board." It then instructs users to refer to the "Getting Started" page for installation instructions. Further down, there's a section about source code development being hosted on GitHub, with links for building the code and verifying it using a PGP key. On the right side, under "DOWNLOAD OPTIONS", there are buttons for Windows (both Win 7 and newer and ZIP file), Linux (32-bit and 64-bit), ARM 32-bit, ARM 64-bit, and Mac OS X (10.10 or newer). Below these are links for Release Notes and Checksums.

Arduino IDE 1.8.16

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

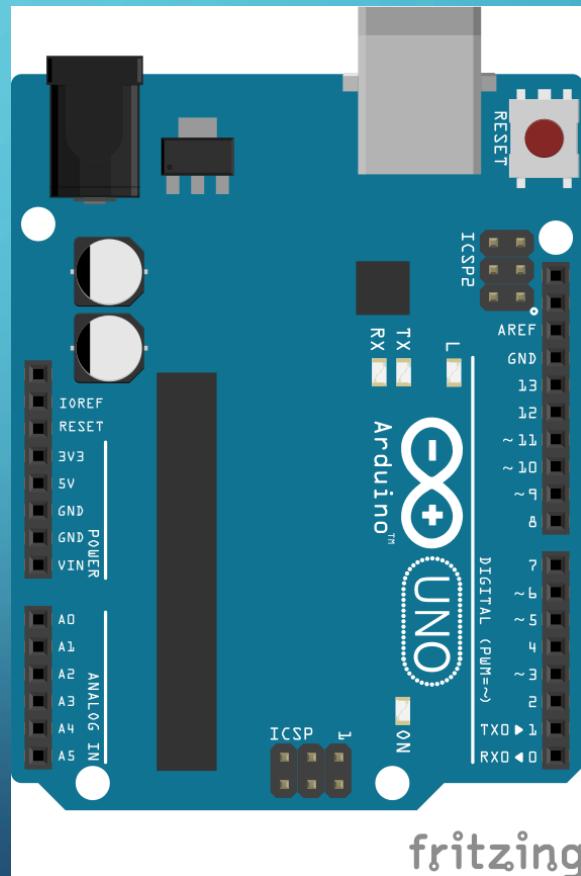
[Release Notes](#) [Checksums \(sha512\)](#)

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

BLINKY EKSEMPEL

EKSEMPEL 1: BLINKY

- Vi tester om Arduinoen virker
- Ved at skrive et "Blinky" program
- Vi skal blot bruge selve Arduinoen
- Vi skriver koden sammen



SENSORER AND AKTUATORER

SENSORER

- Komponenter der er input for Arduinoen
- SENSE som i at mærke omverdenen
 - E.g.:
 - Knapper
 - LDR (Light Dependent Resistors)
 - Potentiometre
- Lette at skrive ind i programmer
 - De elektriske signaler oversættes let til data



SENSORER

- Nogle mere komplikerede sensorer
 - Kommer ofte som "moduler"
 - E.g.:
 - Temperatur-/fugtighedssensor
 - Accelerometer/gyroskop
 - Ultralyds afstandssensor
 - Abstraktion – modulerne indeholder "glue electronics"
- Kommunikerer oftest gennem standard protokoller:
 - I2C
 - SPI
 - UART
- Nogen bruger dog deres egen unikke protokol



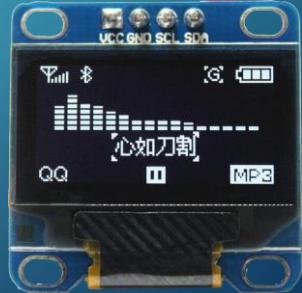
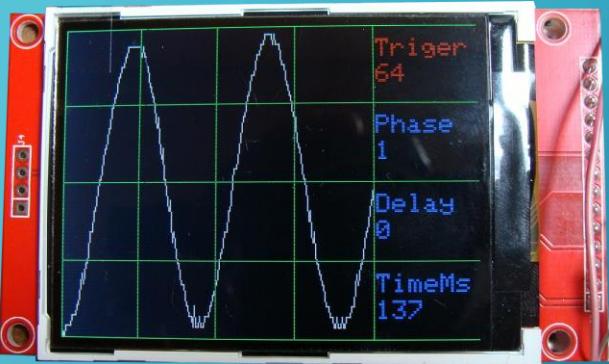
AKTUATORER

- Modpart til sensorer
- Aktuatorer agerer output
- ACT som i at handle på omverden
- De fleste er simple:
 - Motorer
 - Højtalere
 - Lineære aktuatorer
 - Stempler
- Nogle er mere komplicerede
 - Servo motorer
 - Stepper motorer
- De fleste bruger ingen, eller simple protokoller



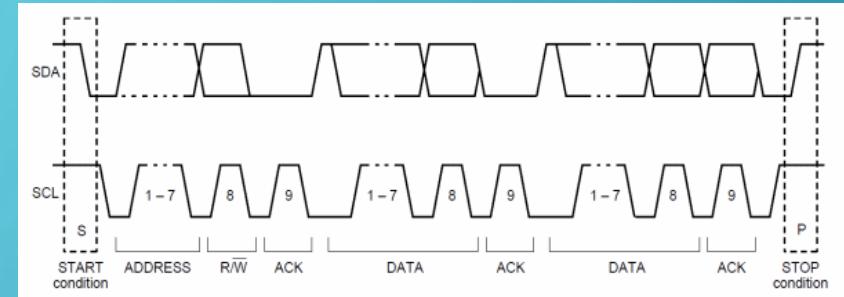
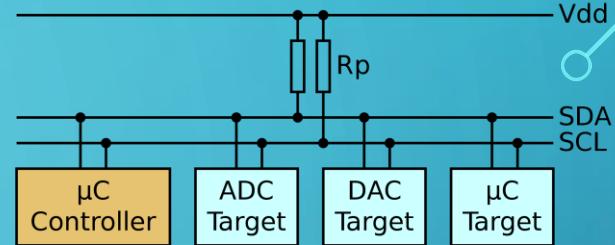
ANDRE OUTPUTS

- Andre outputs tæller ofte lysindikatorer eller interfaces:
 - LED's
 - Tekstdisplays
 - OLED / TFT LCD
- Kræver ofte en standard protokol:
 - SPI
 - Parallel Interface
 - I2C

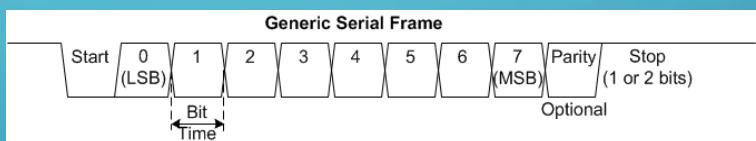
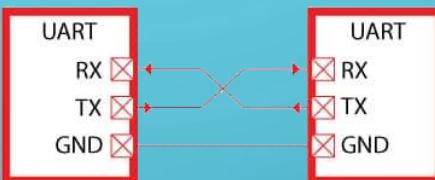


PROTOKOLLER

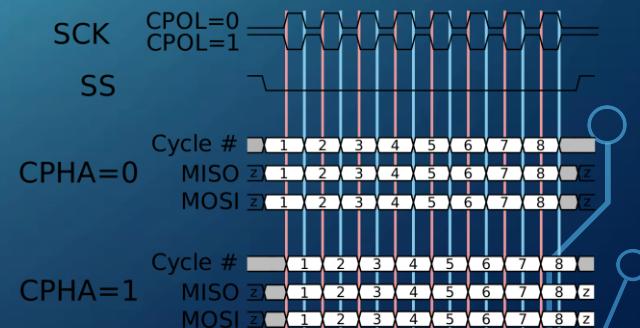
- Protokoller dækker interfaces og "sprogene" mellem computerchips
 - E.g. mellem Arduino og en sensor
- De mest udbredte:
 - UART
 - I2C
 - SPI
- Arduinos libraries håndterer oftest disse for os!



Pictured: I2C connections, and dataframe
URL: <https://en.wikipedia.org/wiki/I%C2%BCC>



Pictured: UART connections, and dataframe
URL: <https://microcontrollerslab.com/uart-communication-working-applications/>



Pictured: SPI connection, and dataframe
URL: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface



DIGITAL AND ANALOG ELECTRONIK

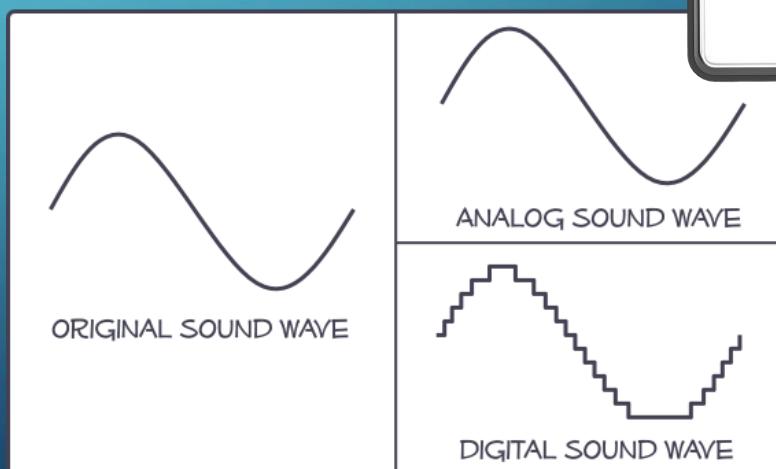
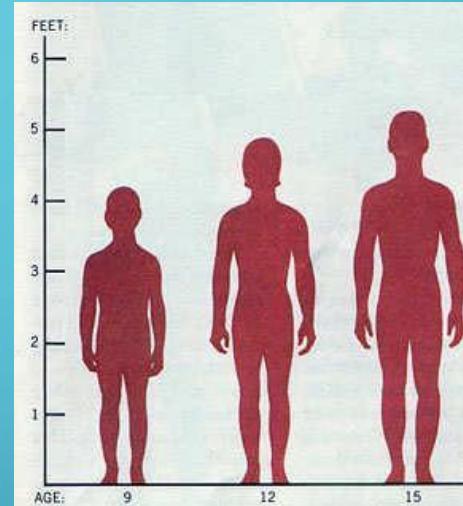
FORSKELLEN MELLEM DIGITAL OG ANALOG?

- Digitale værdier er enten true/false:
 - Er døren åben/lukket?
 - Er knappen trykket ned?
 - Er kaffen klar?
- Også kendt som boolske værdier
(boolean)



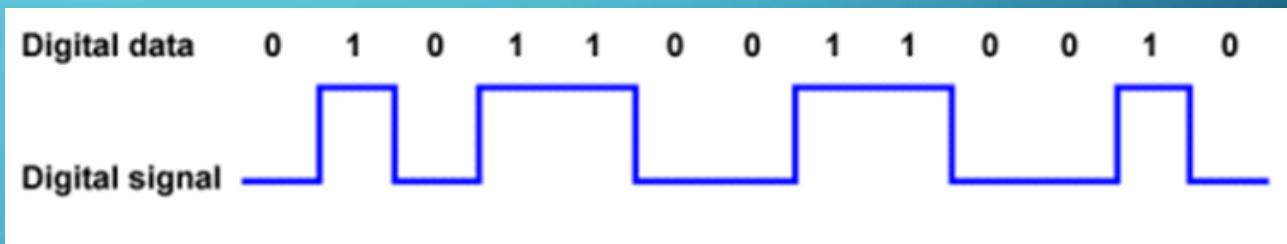
FORSKELLEN MELLEM DIGITAL OG ANALOG?

- Det meste af verden er analog:
 - Folk kan variere i højde
 - Tid kan måles i infinitesimalt små enheder
 - Lige som andre fysiske fænomener:
 - Indenfor HiFi audio, er der meget påstyr over små detaljer i lydbølger



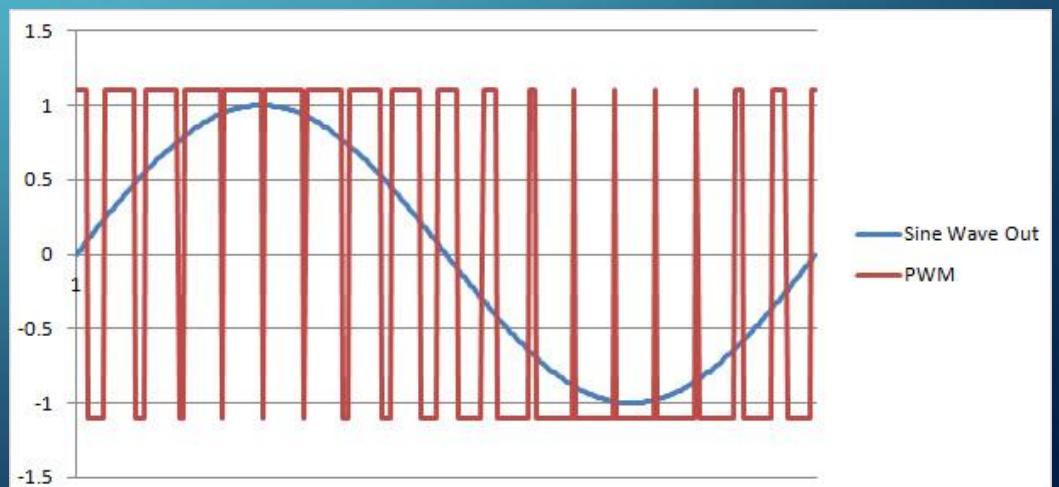
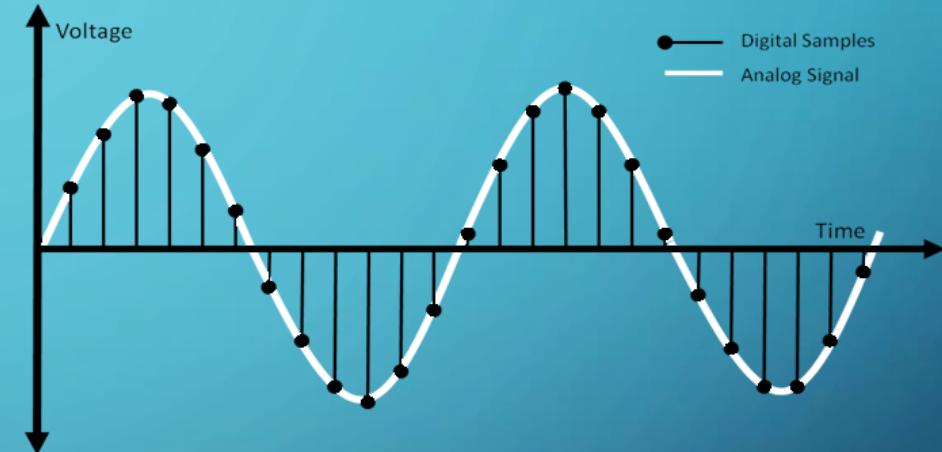
DIGITAL PÅ ARDUINO

- Arduino kan "read" og "write" digital data:
 - Som en spænding, enten 0V eller 5V
 - 0V = 0/false/LOW
 - 5V = 1/true/HIGH
- Repræsenteres på Arduino som enten: HIGH eller LOW
- Spændinger bestemmes udfra en tærskel, hvor de tælles som enten 1 eller 0 i data



ANALOG PÅ ARDUINO

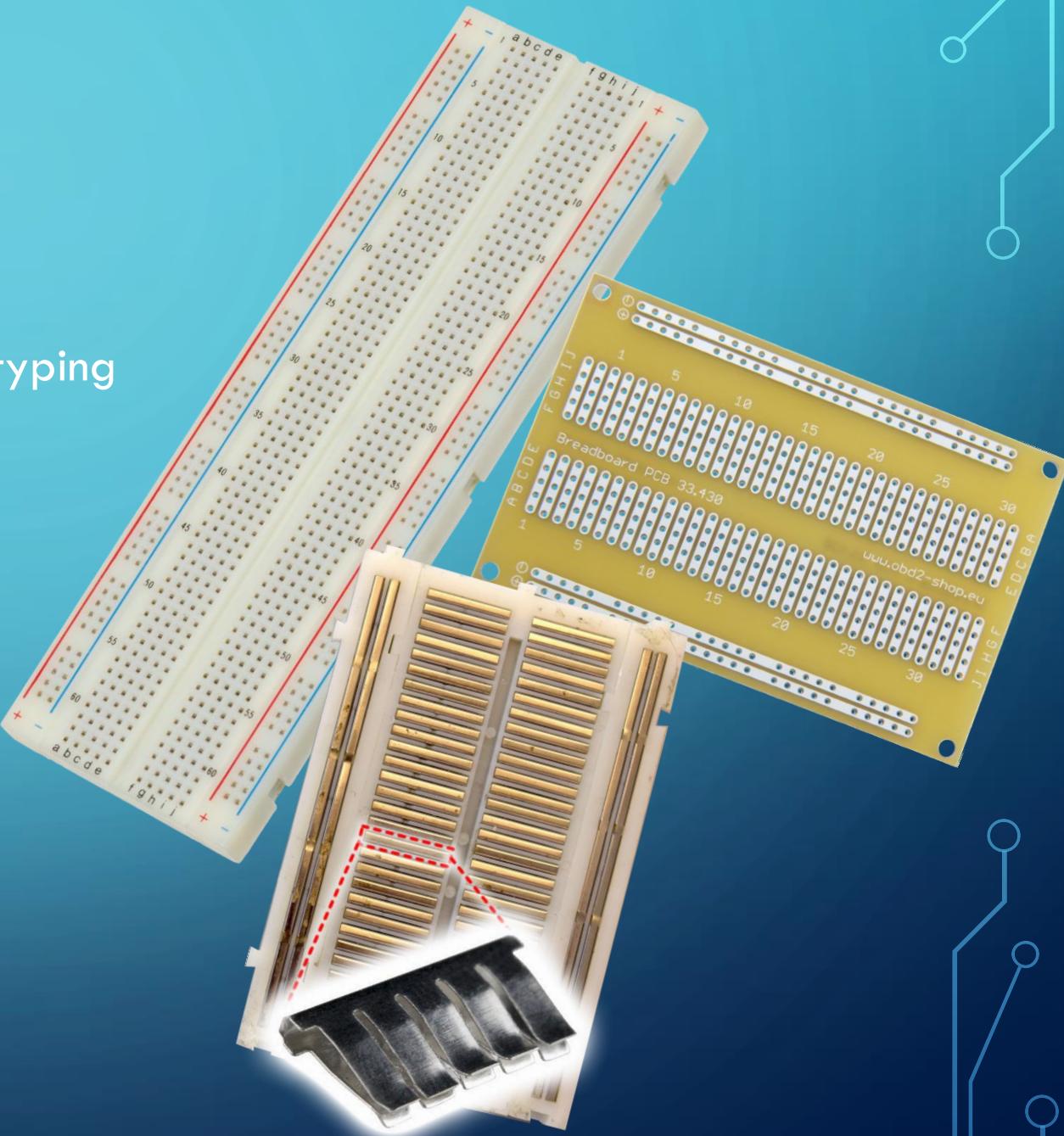
- Arduino kan også måle analog data
 - Gøres på dens analog input pins
 - Kaldes "sampling"
- Den kan også outputte et "semi" analogt signal
 - Gøres vha. PWM (Pulse Width Modulation)
 - Hurtig tænd/sluk for et digitalt output
 - Samme metode som class D forstærker
 - Eller nogle typer spændingsregulator (DC to DC)



EKSEMPLER

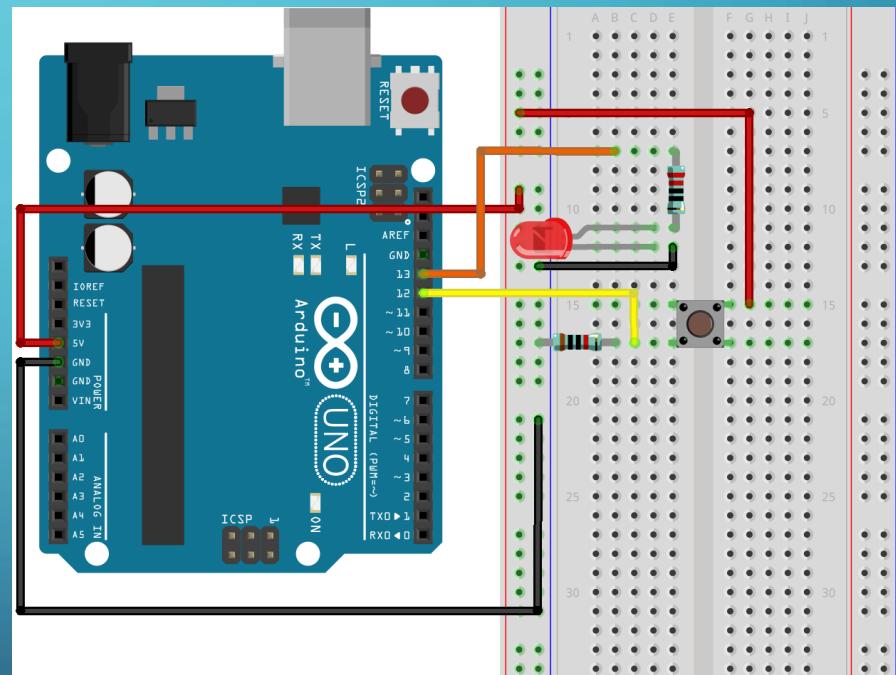
OM BREADBOARDS

- Breadboards er til for at tillade let prototyping af elektronik
- Ledninger sættes i hullerne, og danner forbindelser gennem metalklemmer på indesiden
- De har to "række" typer af klemmer:
 - Bus strips
 - Forbundet hele vejen
 - Terminal strips
 - Forbinder ~5 pins
 - Afbrudt på midten



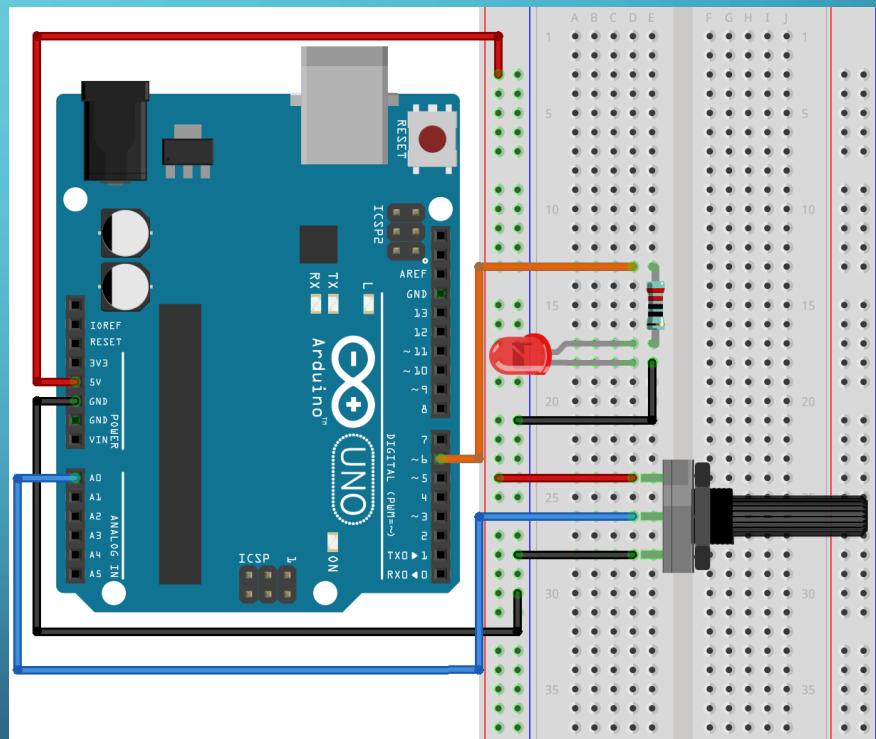
EKSEMPEL 2: BUTTON

- Nu vil vi bygge et kredsløb
- Vi tilføjer et input – en trykknap
- LED'en bliver styret gennem en boolsk værdi



EKSEMPEL 3: ANALOG

- Vi prøver analog sampling!
 - Vi bygger et nyt kredsløb
 - Og får Arduino til at kommunikere med PC'en
 - Vi bruger PWM ("analog output") til at regulere LED'en



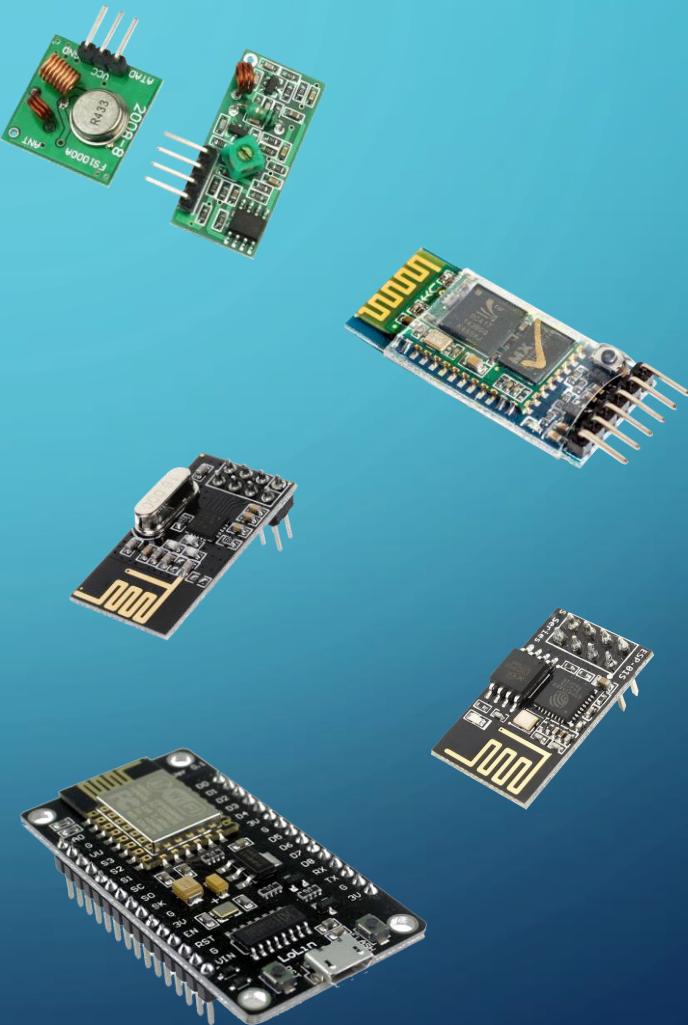


DEL 2: WIRELESS COMMUNICATION



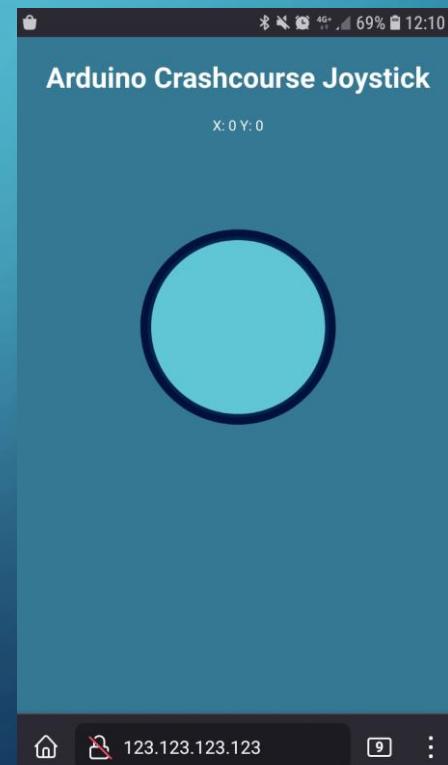
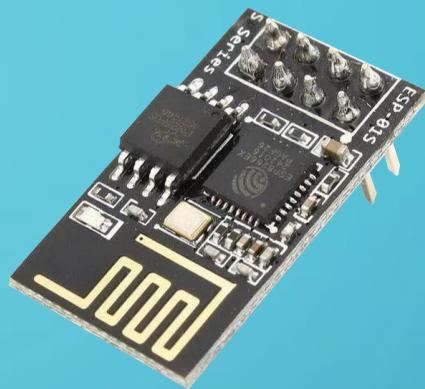
TYPER AF TRÅDLØS KOMMUNIKATION

- FS1000A + XY-MK-5V
 - 433MHz - envejs
 - Amplitude Shift Keying
 - Meget følsom of verfor støj
- HC-05
 - 2.4GHz Bluetooth 2.0
 - Fungerer som både master / slave
- NRF24L01
 - 2.4GHz
 - Gaussian Frequency Shift Keying
 - Understøtter BLE og "Enhanced Shockburst"
- ESP-01 / ESP-8266
 - WiFi -2.4GHz
 - Fungerer som Access Point eller Client
- ESP-32
 - Beefed up ESP-8266 med BLE oveni
- ESP's kan også bruges som MCU'er i sig selv



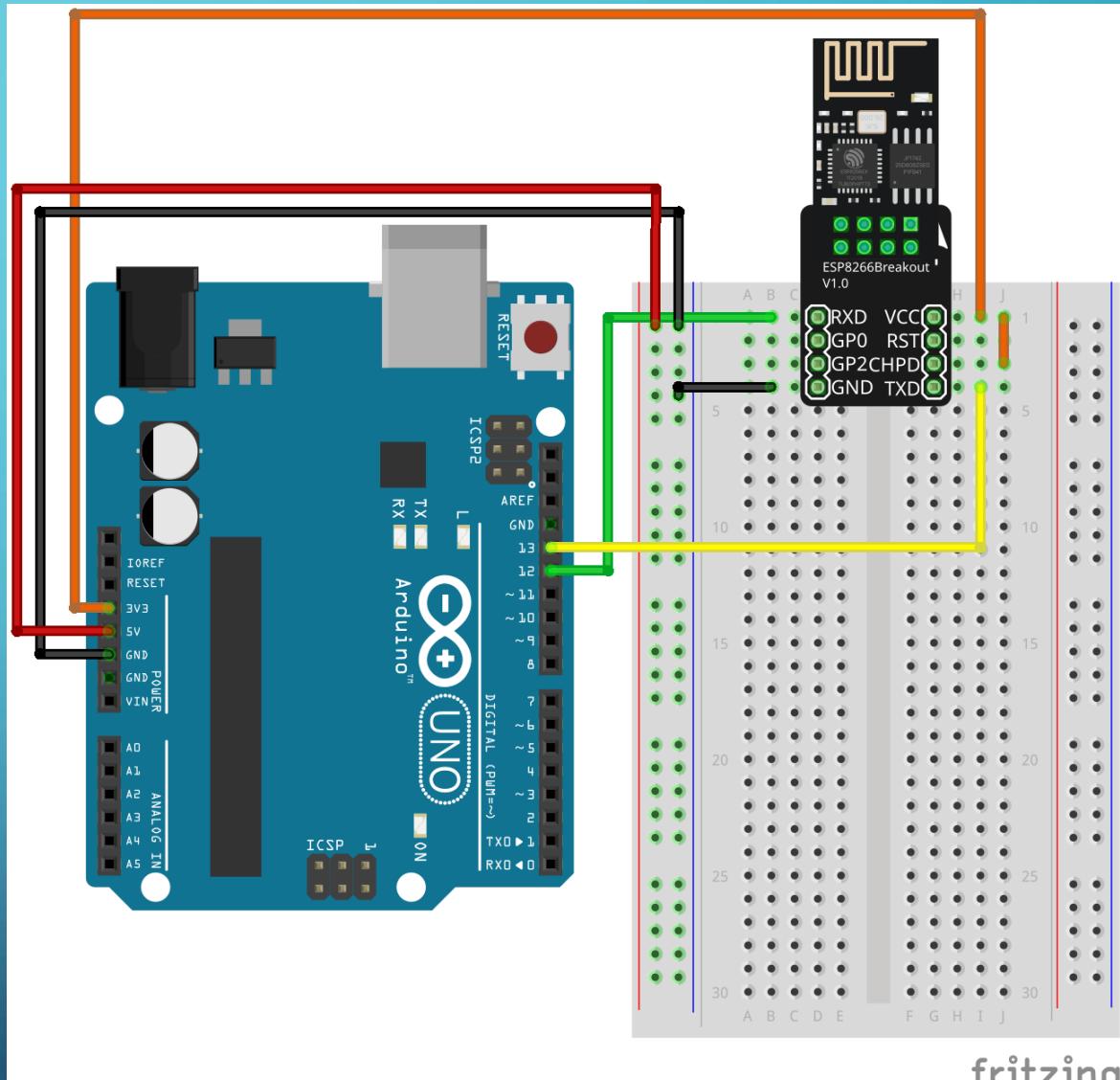
ESP-01

- ESP-01 indeholder en ESP8866 MCU
- Er sat op som Serial "Modem"
 - Giver Arduino en (meget) simpel WiFi forbindelse
- Velegnet til simple IoT projekter
- Vores har hjemmelavet software (jeg har skrevet det)
 - Tilgængelig:
<https://github.com/iakop/ArduinoCrashcourse>
- Serverer en side gennem custom access point og IP
 - Sender X og Y koordinater over Serial til Arduino
 - Kan bruges til fjernstyring



WIRELESS EKSEMPEL

- The ESP-01 bruger et Breadboard breakout
 - Vi forbinder den til Arduino gennem Software Serial
 - Arduino giver den en IP, SSID and password
 - For nu skal vi bare tilgå dens hjemmeside
 - Og sende/aflæse X og Y over Serial





DEL 3: MOTOR CONTROL



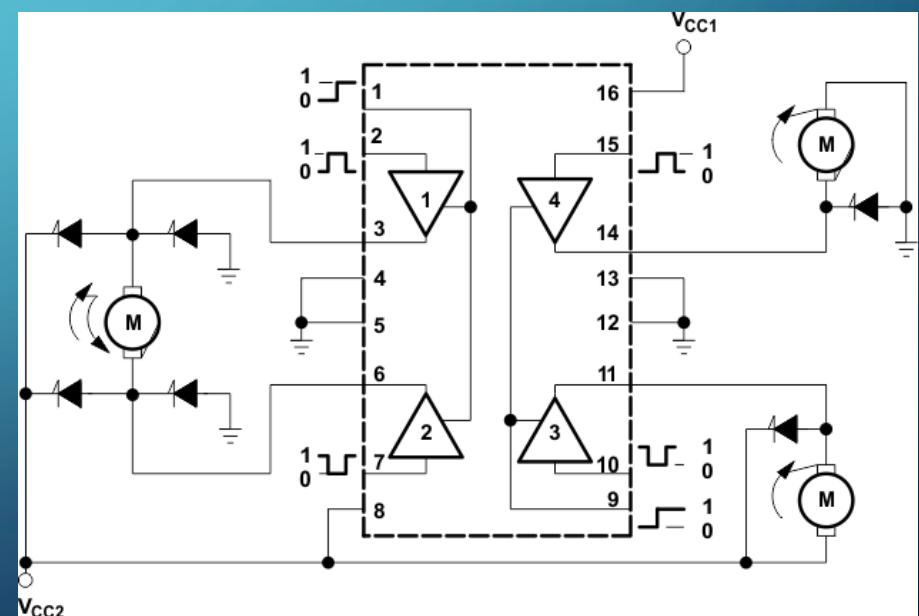
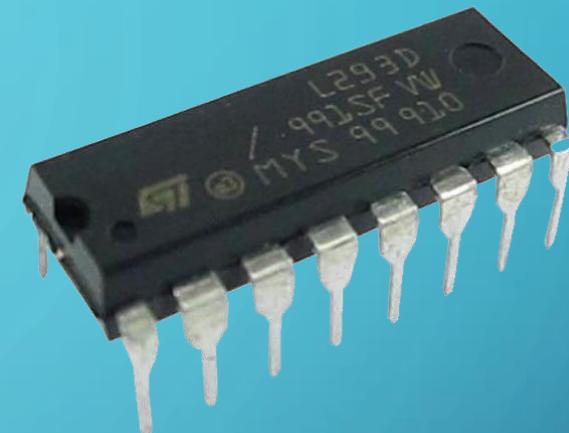
DC MOTORER

- DC motorer er de simpleste motorer at styre
- Kræver blot en styrestørrelse
- Kan variere hastighed ved hurtig / tænd sluk
 - Oftest vha. PWM
 - "analog hastighed"
- Retning styres vha. polariseringen



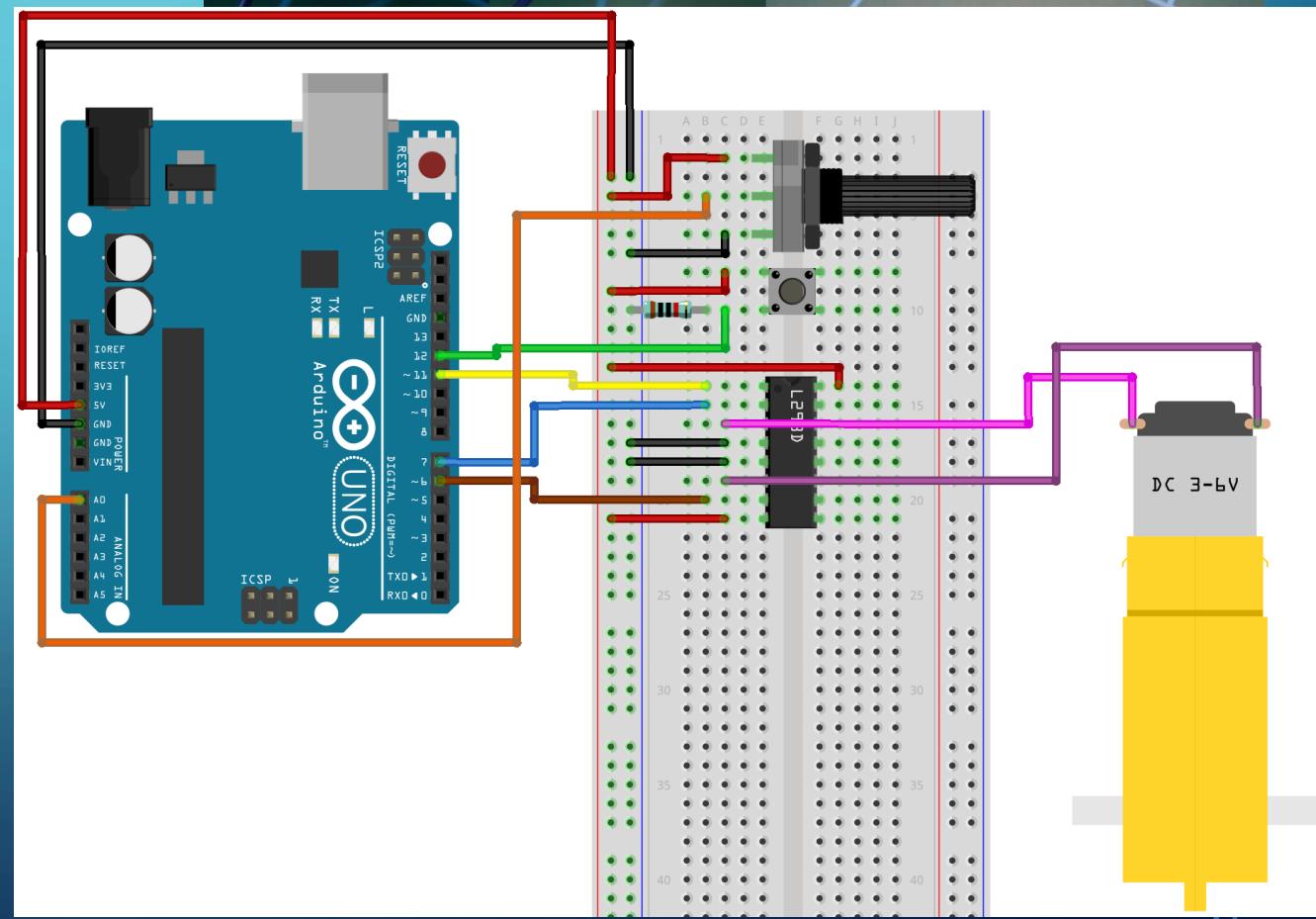
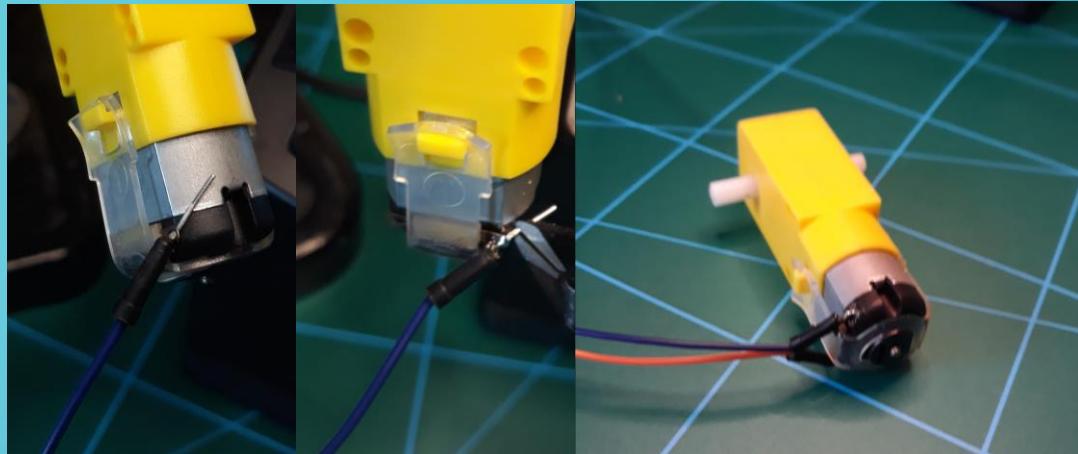
L293D DRIVER

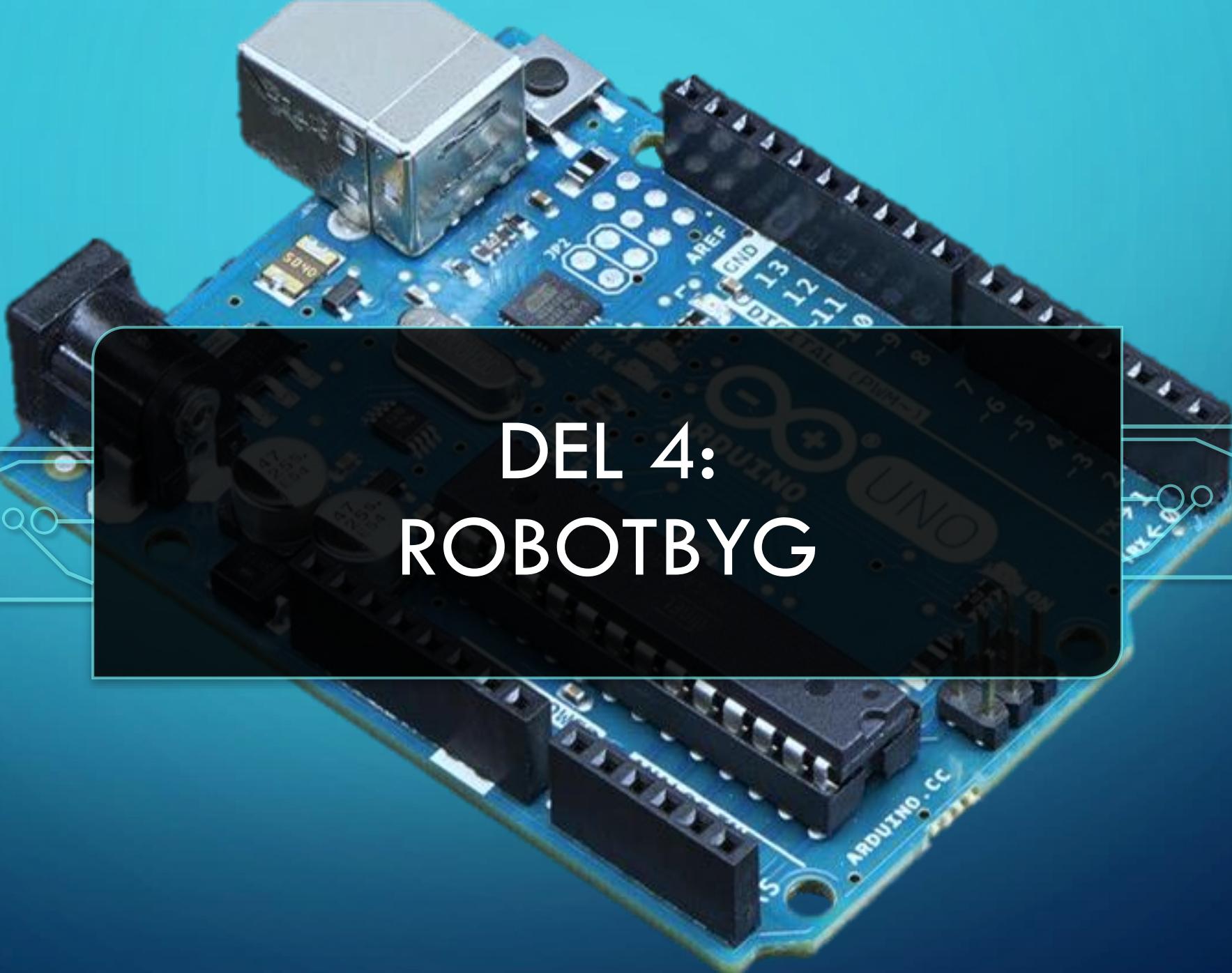
- Arduino Kan ikke outputte nok strøm til at køre en motor
 - Den behøver en "forstærker", eller driver
- L293D er en quad half H-bridge driver
- Kan bruges til:
 - 4 motorer one way
 - 2 motorer back/forth
 - Vi skal bruge 2 back/forth



MOTOR DRIVING EKSEMPEL

- Vi tilslutter Arduino kontrol pins'ne på L293D
 - Sæt en DC-motor fra sættet i
 - Det skal først loddes
 - Placér et kabel i hullet, og lod det
 - Der skal være to kabler loddet
 - Vi kan ordne begge motorer nu vi er igang
 - Vi programmerer Arduino til at styre den
 - Regulerer hastigheden med analogWrite() og et potmeter





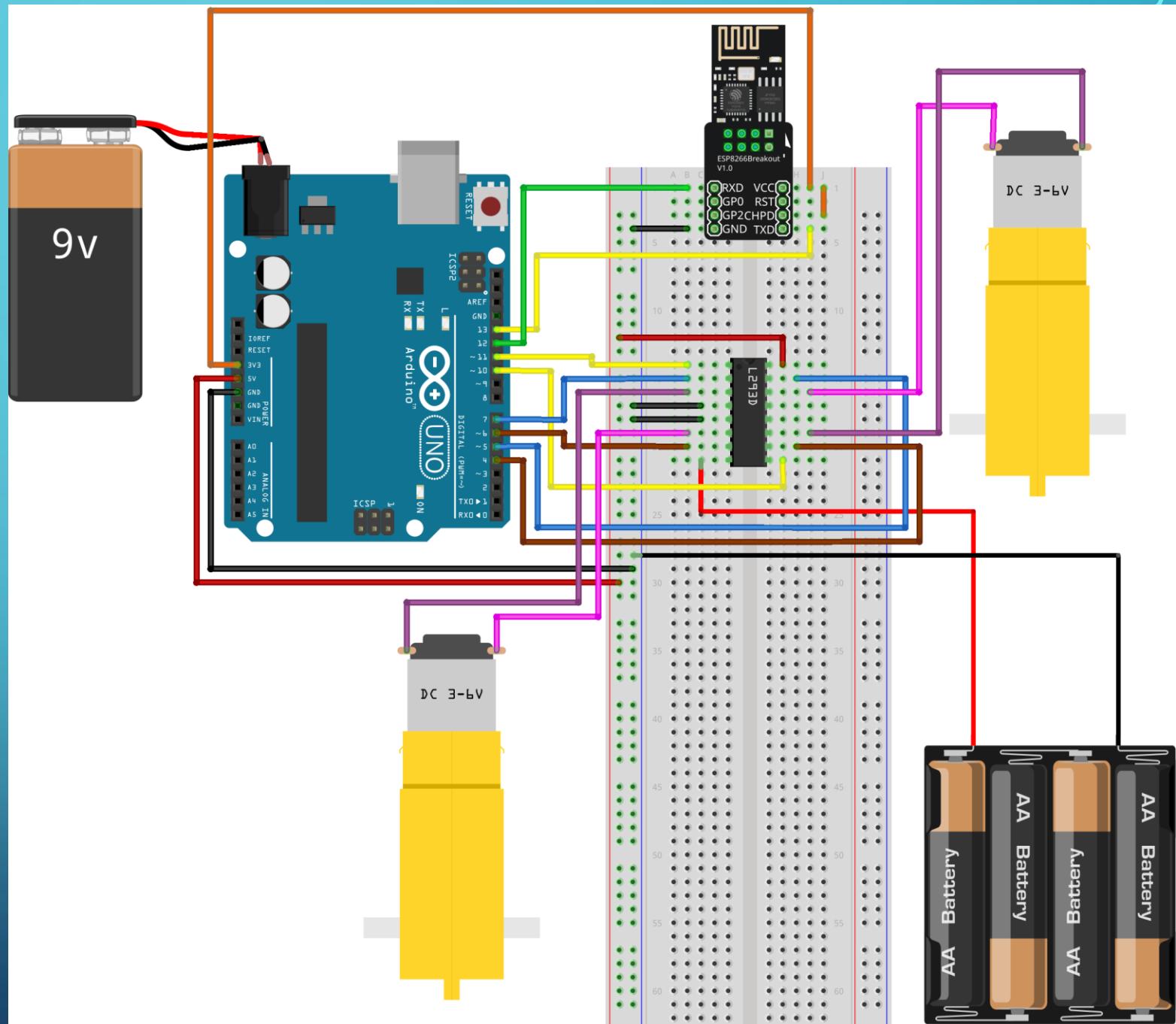
A close-up photograph of an Arduino Uno microcontroller board. The board is blue with various electronic components, including a central ATmega328P microchip, resistors, capacitors, and a USB port. It features a 2x14 pin header for digital pins and a 16-pin analog pin header. The word "ARDUINO" is printed on the board, along with the model name "UNO". A black rectangular overlay box is centered on the board, containing the text "DEL 4: ROBOTBYG".

DEL 4: ROBOTBYG



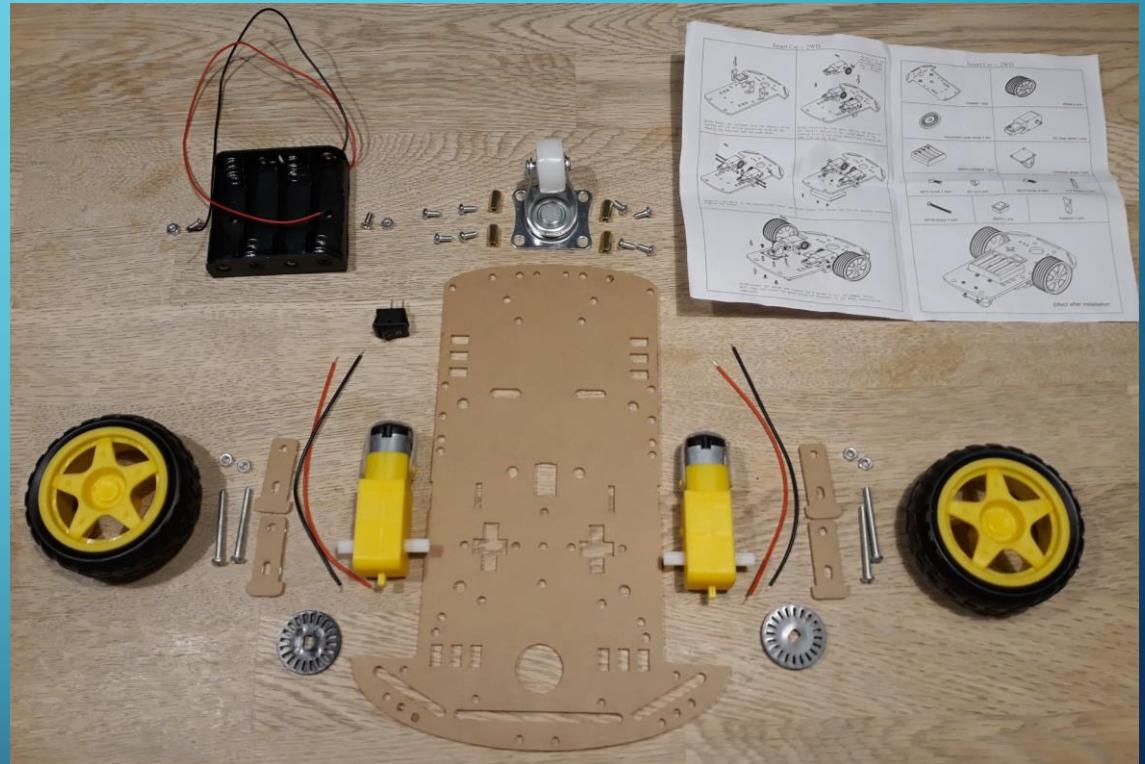
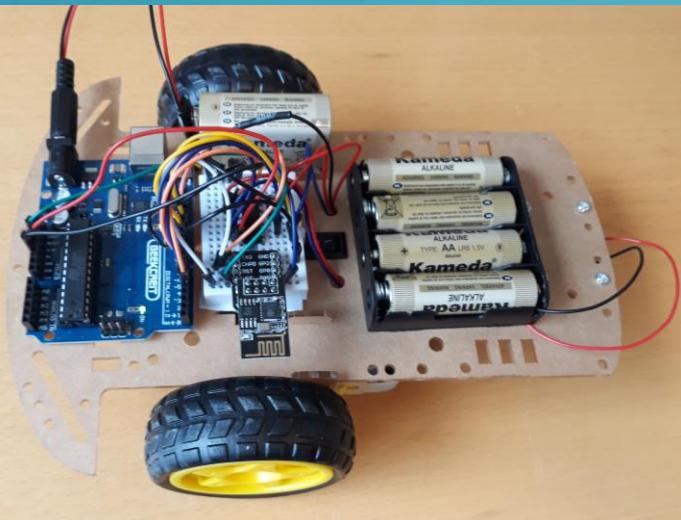
PUTTING IT ALL TOGETHER

- Vi kan nu sætte motorstyring sammen med ESP-01
- Two DC motorer går i L293D
- ESP-01 forbindes til Arduino
- Arduino skal oversætte koordinaterne til motorstyring



GENNEMGANG AF ROBOT

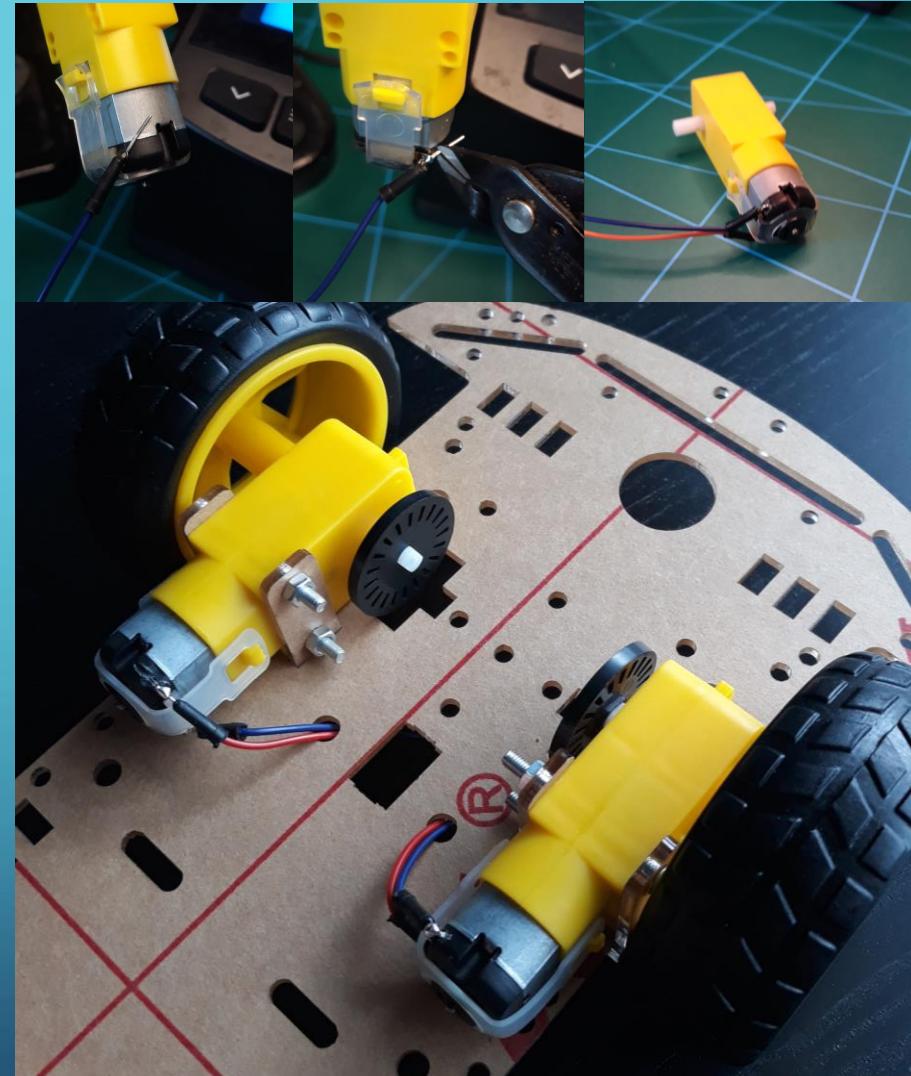
- I har modtaget et robotsæt
 - Instruktioner til samling er inkluderet
- Når det er samlet kan I sætte krdsløbet på
 - Her er et eksempel



- For det lille breadboard, er et eksempel på github

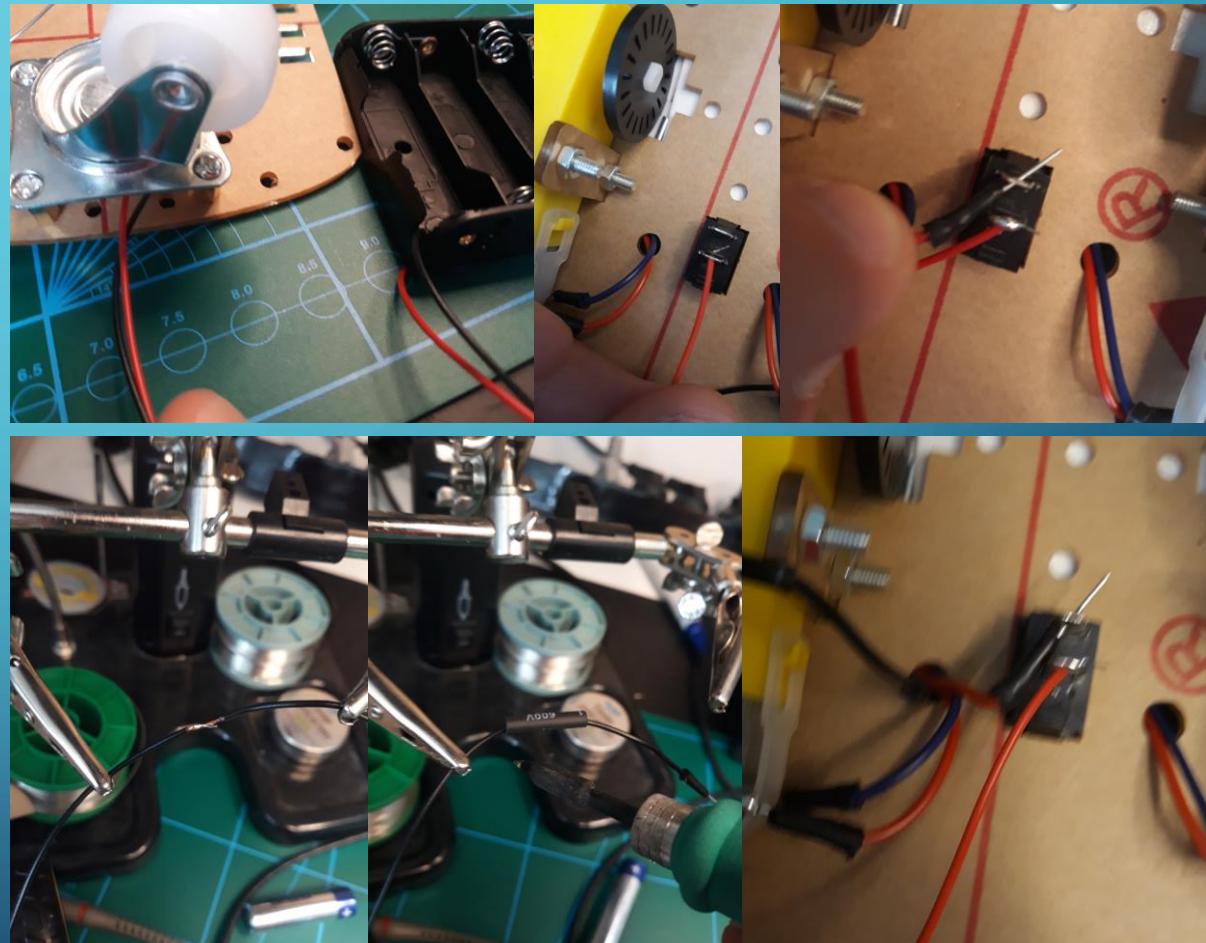
GENNEMGANG AF ROBOT

- Husk at lodde
 - Begge motorer
 - Placer kabler i hullerne, og lod dem
 - Det skal monteres på undersiden, og kablerne kan trækkes op igennem pladen



GENNEMGANG AF ROBOT

- Husk at lodde
 - Batteriholder
 - Kablerne føres gennem bagsiden af robotten
 - + (RØD) Skal forbindes til den ene ende af den monterede powerknap
 - Endnu en + (RØD) ledning loddes på den anden ende af powerknappen
 - – (SORT) Forlænges med en ekstra ledning
 - Krympeflex kan bruges for sikkerhed
 - Kablerne køre gennem huller til oversiden
 - Nu kan de levere strøm gennem L293D



FREESTYLE BYG

- Resten af byg kan gøres efter eget tempo og behov
- Vi har loddestationer, skævbiddere, skruetrækere osv.
 - Bare deles påent om dem
- Jeg er tilgængelig for eventuelle spørgsmål
 - Kan også hjælpe med eventuelle ekstra tilføjelser til jeres kredsløb
 - Men hold det simpelt og muligt!

AFRUNDING

- Nu kender I til de basale principper
- I har prøvet at bygge en simpel robot
- Forslag til tilføjelser I kan eksperimentere med derhjemme:
 - Ultralydssensor
 - Servo motor arm (servo er i jeres kit)
 - Flere knapper på web interface
 - Anderledes kontrol-opførsel
- Mange moderne projekter ligger på internettet
 - ESP boards er her for at blive
 - Kan også programmeres gennem Arduino IDE

EVALUERING

- Jeg vil meget gerne have direkte feedback
 - Var det noget i syntes godt om, noget ikke?
 - Kunne noget formuleres bedre?
 - Hvordan føltes tiden ift. materialet?
 - Er der tid nok?
- Men mest af alt TAK for jeres deltagelse!

RESURSER

- Nogle online resurser I kan bruge:
 - Slides, eksempler, osv.
 - <https://github.com/iakop/ArduinoCrashcourse>
 - Arduino IDE Download
 - <https://www.arduino.cc/en/software>
 - Arduino Language Reference
 - <https://www.arduino.cc/reference/en/>