

A blue Arduino Uno microcontroller board is shown from a top-down perspective, angled slightly. A black USB cable is connected to the board's USB port. The board features a central ATmega328P microcontroller chip, various surface-mount components like resistors and capacitors, and a breadboard-style pin header. A dark rectangular overlay with rounded corners is centered on the board. Inside this overlay, the text 'ARDUINO CRASHCOURSE' is written in large, white, sans-serif capital letters, followed by a smaller line of text '- ZERO TO HERO!' in the same style.

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 årige "Inventors"
- Afholdt kurser i Arduino for IDA, StudyNow, Herning Kommune, etc.
- Embedded electronics engineer på DTU
 - Robotter og autonome systemer

FORMÅL MED DENNE WORKSHOP

- At give en HURTIG introduktion til de essentielle dele af Arduino:
 - Noget hands-on succes
 - Møder med nogle af udfordringerne
- Inspiration til hvordan I selv kan bruge Arduino
- Snuse til IoT mulighederne ved ESP32 på Arduino platformen

FØR VI STARTER

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

AGENDA

- 16:00-17:30 : Arduino Basics
- 17:30-18:00 : Setup af ESP32 til Arduino
- 18:00-18:30 : Aftensmad - Pizza og sodavand
- 18:30-19:45 : IoT på ESP32
- 19:45-20:00 : Afrunding og tak for denne gang!

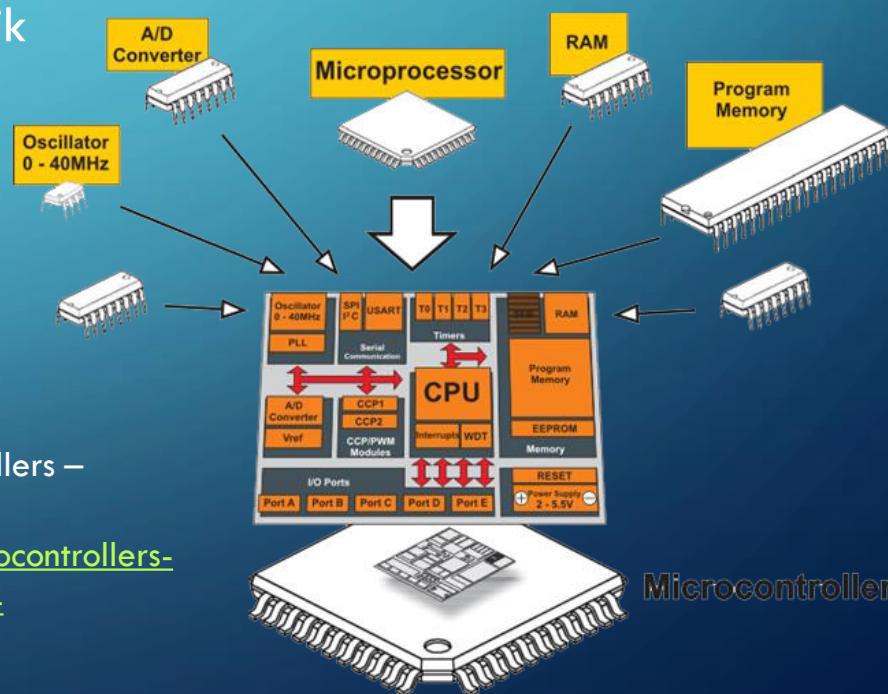


ARDUINO BASICS



HVAD ER ARDUINO?

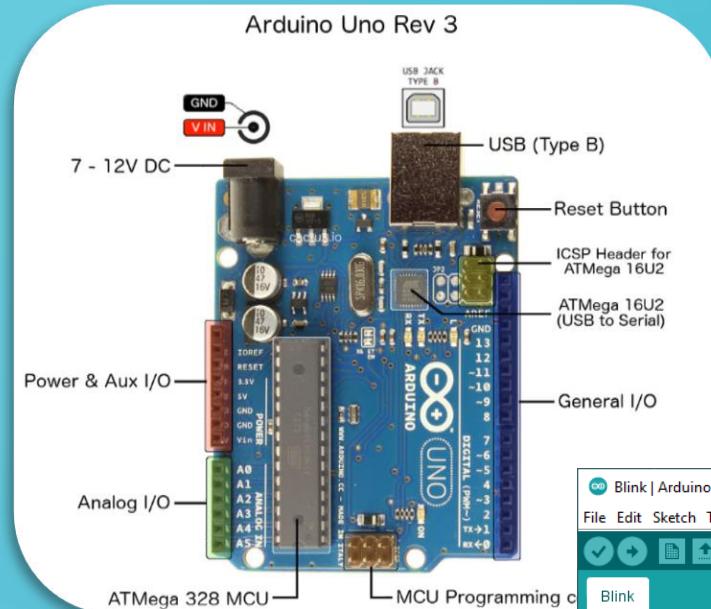
- Mest brugte platform til MCU prototyping
 - MCU står for Microcontroller Unit
 - Et lille computersystem, der styrer anden elektronik
- Simpel, intuitiv programmering
- Mange eksempler på kode tilgængelige
 - Mange software libraries



Figur: 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

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



Arduino og dens pins

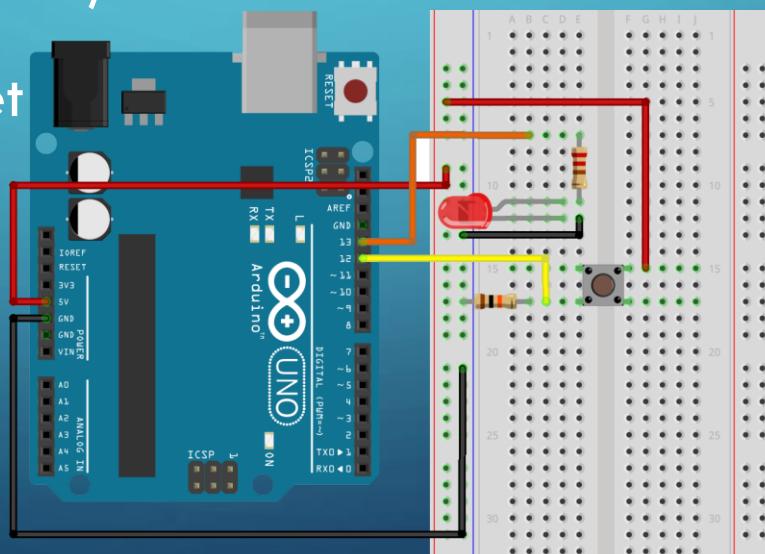
The screenshot shows the Arduino IDE interface with the following details:

- Title bar: Blink | Arduino 1.8.15
- Menu bar: File Edit Sketch Tools Help
- Sketch window:

```
// 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
}
```
- Status bar: Arduino Uno on COM4

Det officielle Blink program i Arduino IDE



Arduino forbundet til elektronik gennem breadboard

HVORFOR ARDUINO?

- Sammenlign disse to programmer
 - Hvilket giver mest mening at læse?

```
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 }
```

```
1 const int ledPin = 13; //Physical LED pin
2 const int btnPin = 3; //Physical button pin
3
4 void setup() {
5     //Initialize 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/ArduinoCrashcourseLoT>

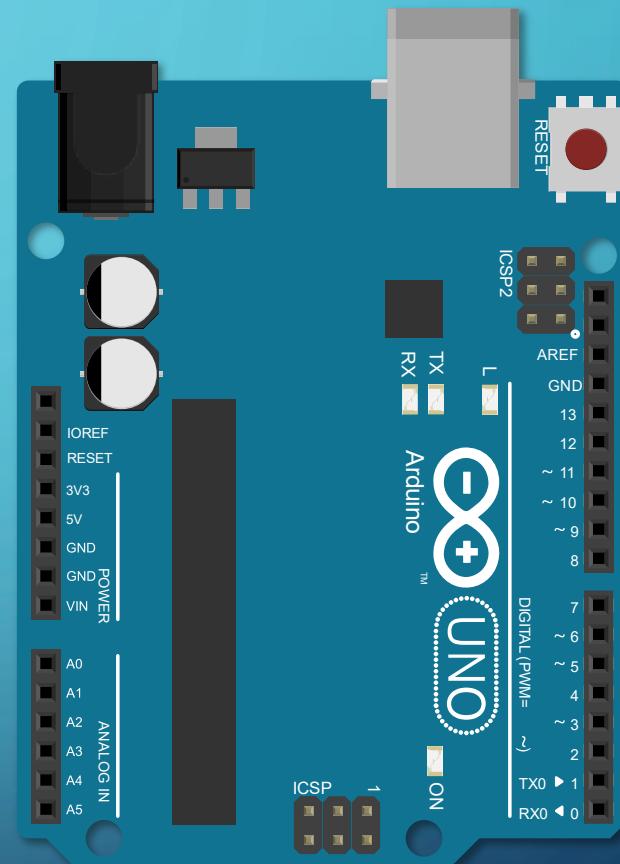


The screenshot shows the Arduino IDE download page. At the top left is the Arduino logo (an infinity symbol with a plus sign). To its right is the text "Arduino IDE 1.8.16". Below the logo is a brief description: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the 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). A red banner at the bottom of this column reads "ALDRIG HENT DENNE HER". Below the download buttons is a link to "Release Notes Checksums (sha512)".

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

EKSEMPEL 0: BLINK

- Vi tester om Arduinoen virker
- Vi kører det officielle Blink eksempel under:
"File > Examples > 01.Basics > Blink"



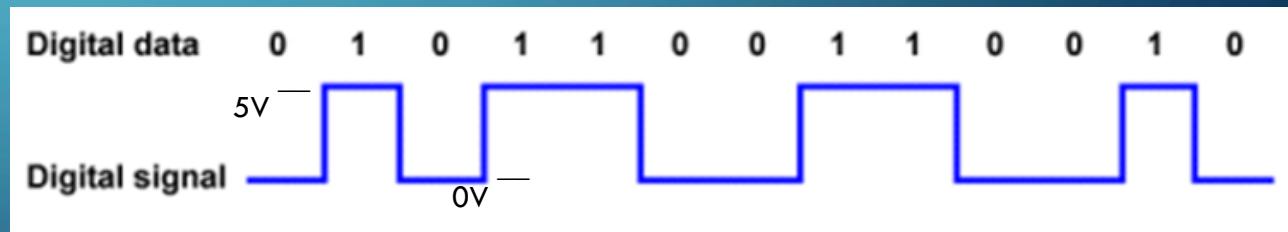
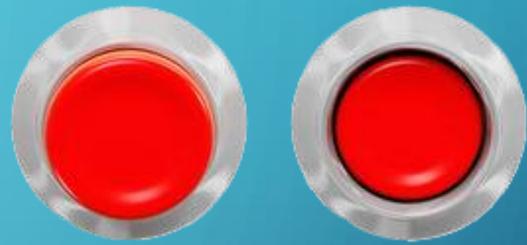
OM DIGITALE KOMPONENTER

- Blink eksemplet brugte en indbygget LED
 - Som digitalt output
- Digitale outputs er f.eks.
 - LED'er
 - Motorer
 - Relæ'er
- Digitale når de udelukkende kan være enten tændte eller slukkede
- Digitale inputs er f.eks.
 - Knapper
 - PIR sensor
 - Tilt sensor



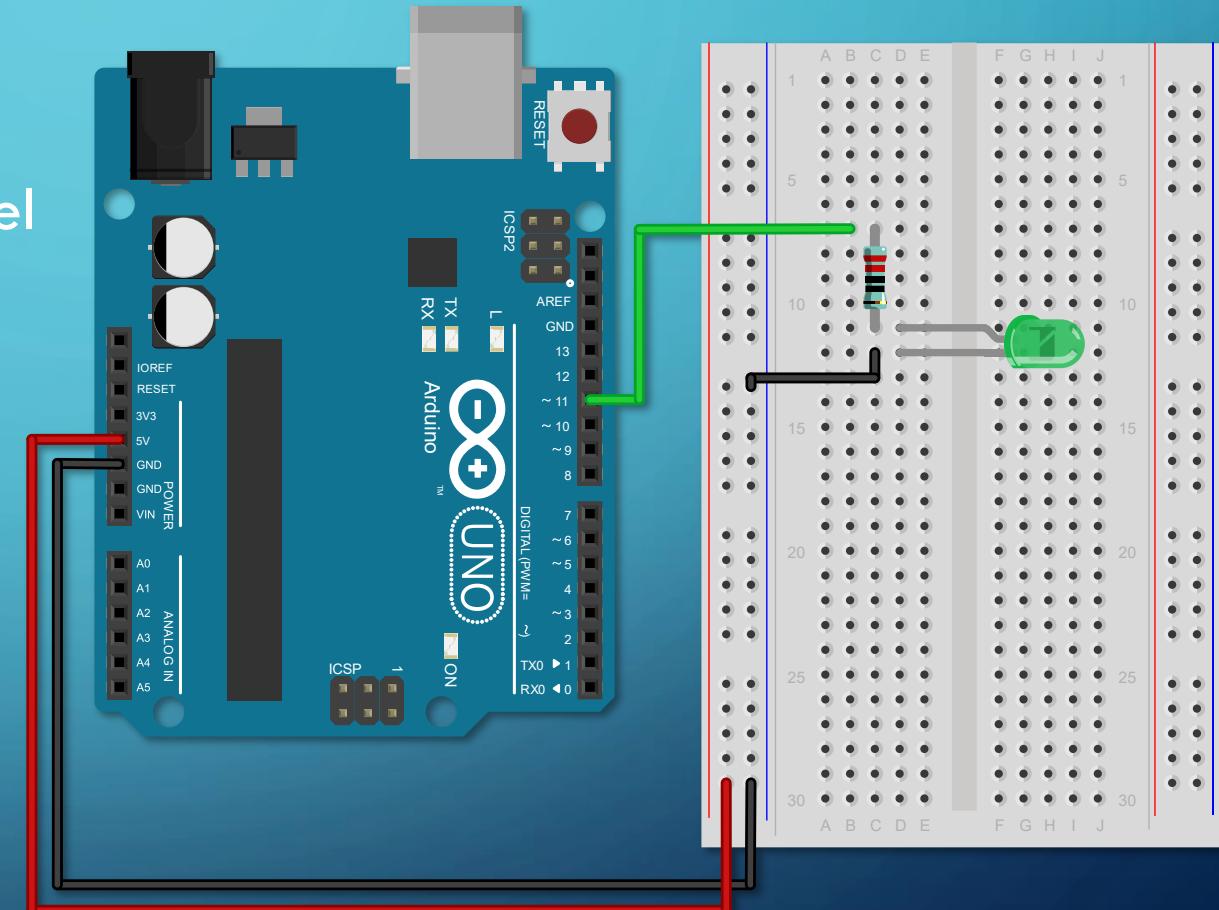
DIGITAL PÅ ARDUINO

- Digitale data er boolske værdier;
 - Enten "True" eller "False"
 - Eksempler:
 - "Er døren åben"
 - "Er knappen trykket"
 - Arduino får dataene fra fysiske spændinger:
 - 5V for True
 - 0V for False
 - Repræsenteret i koden som HIGH eller LOW



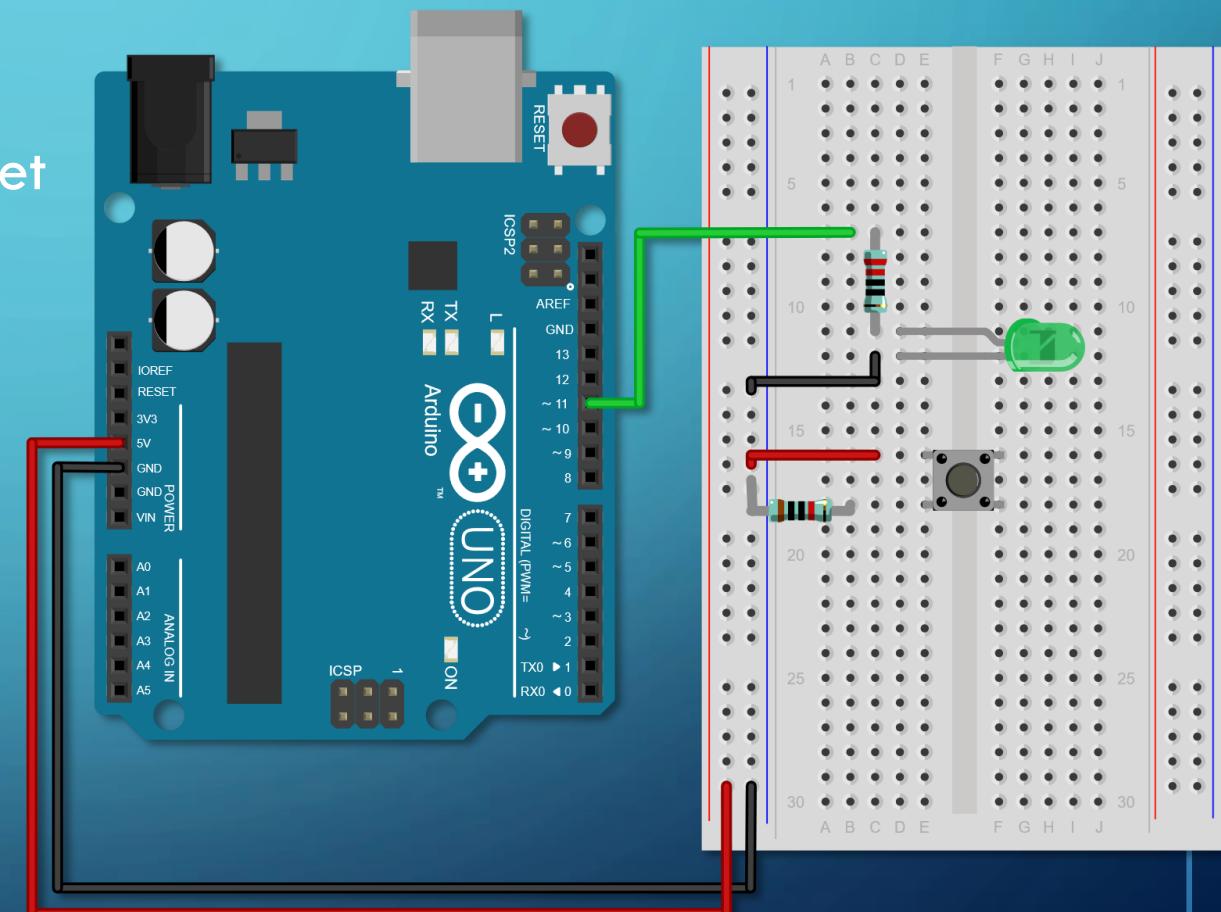
EKSEMPEL 1: BLINKY

- Vi bygger nu selv et blink eksempel
- Det gøres med eget kredsløb
- Koden skriver vi også selv



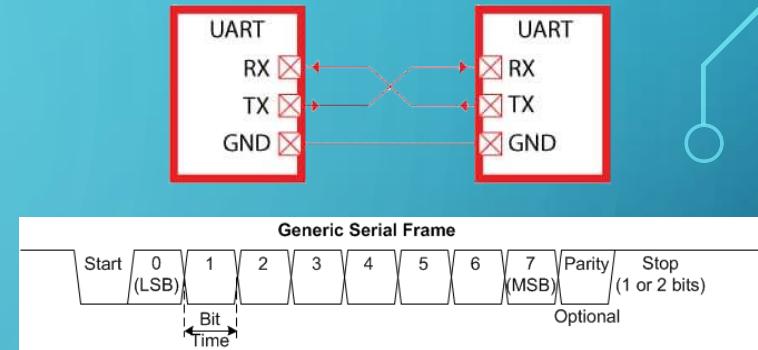
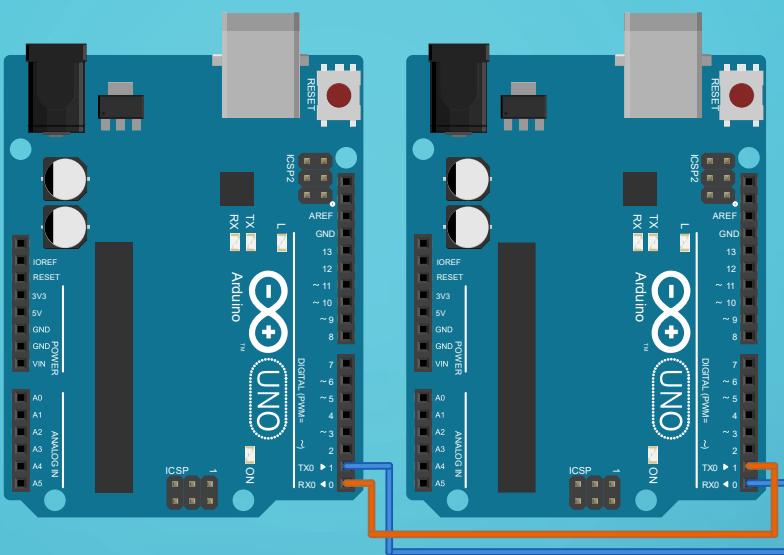
EKSEMPEL 2: BUTTON DEBOUNCE

- Vi sætter en knap til Blinky eksemplet
- Den skal få LED'en til at tænde og slukke
- Vi skal behandle nogle mekaniske fænomener i koden

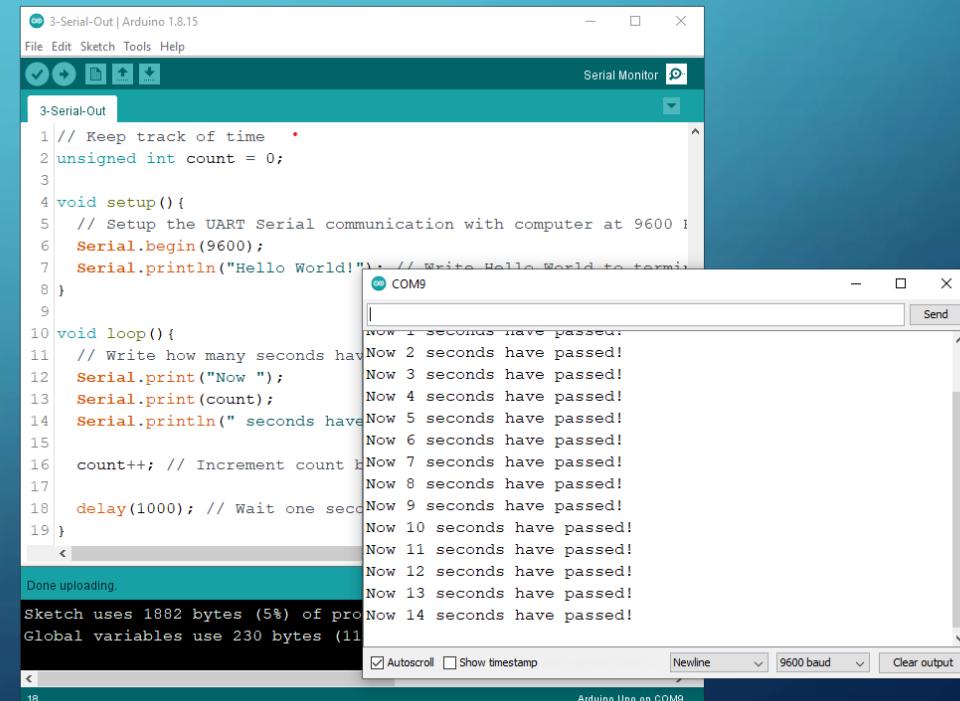


SERIAL - UART

- Serial er kommunikation gennem en serie af tegn
 - I modsætning til parallel
- Arduino snakker med computeren gennem Serial
 - Mere specifikt UART
 - Universal Asynchronous Receiver/Transmitter
 - Data sendes mellem TX/RX pins gennem en converter over USB

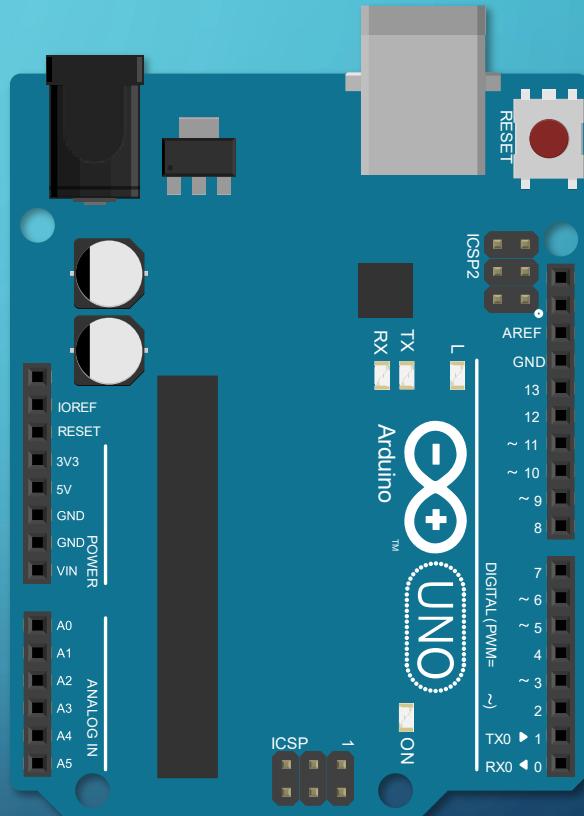


Billede: UART forbindelse og dataframe
URL: <https://microcontrollerslab.com/uart-communication-working-applications/>



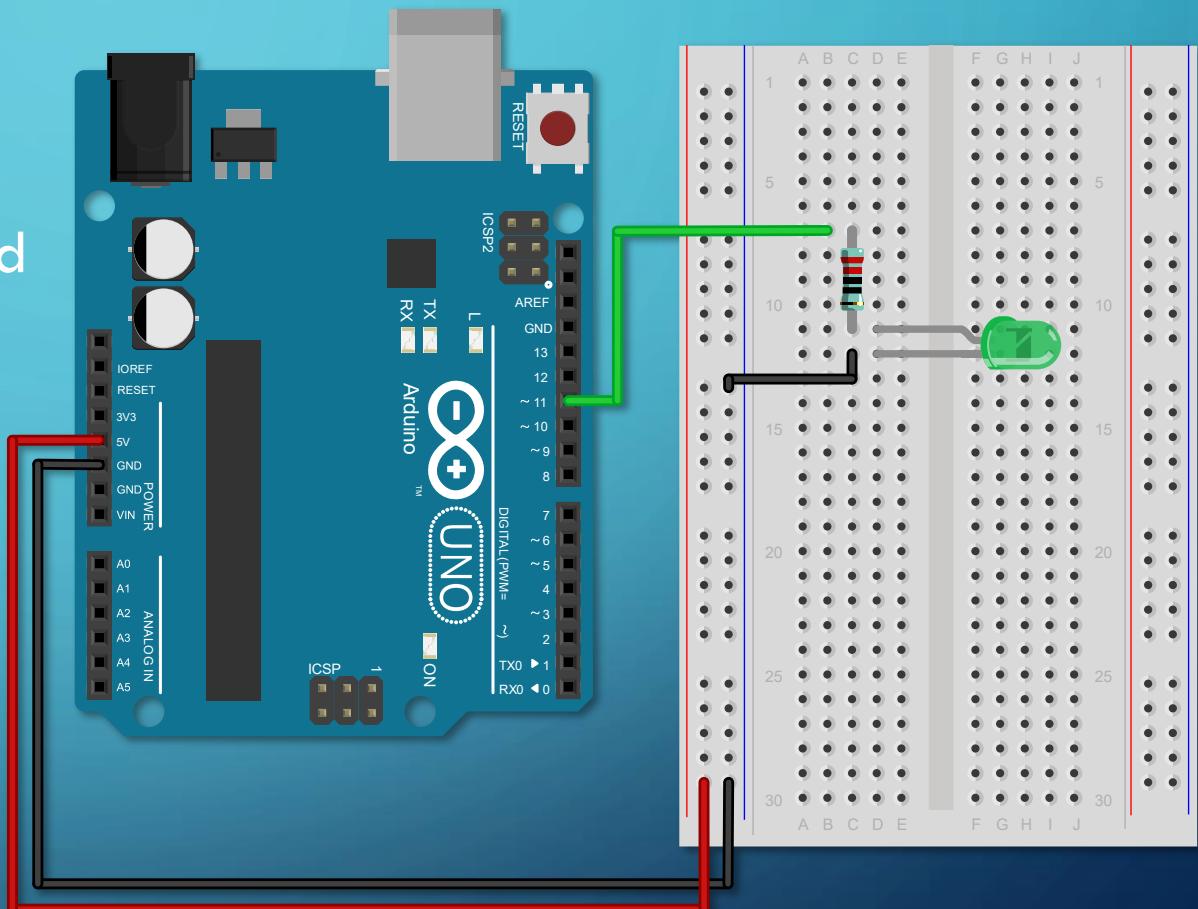
EKSEMPEL 3: SERIAL OUT

- Arduinoen skal skrive en besked til computeren
- Serial Monitor findes i "Tools>Serial Monitor"



EKSEMPEL 4: SERIAL IN

- Arduinoen skal modtage en besked fra computeren
- I Serial Monitor skal vi kunne tænde/slukke en LED



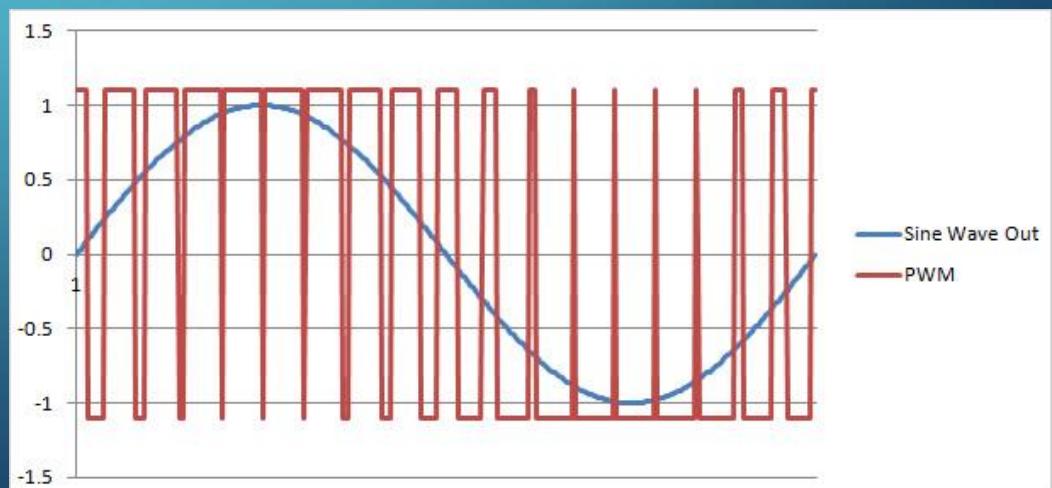
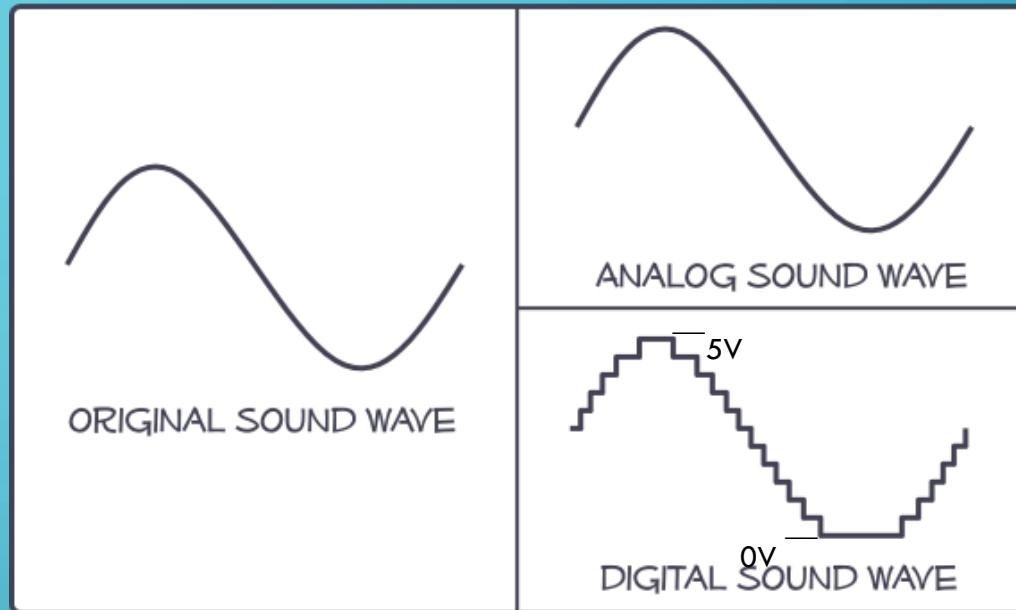
OM ANALOGE KOMPONENTER

- Analoge output og input er ikke kun True eller False
- Analoge outputs er f.eks.:
 - LED (Variabel lysstyrke)
 - Motor (Variabel hastighed)
 - Servomotor
- Analoge input er f.eks.:
 - Potentiometer
 - Joystick
 - Lysfølsom modstand



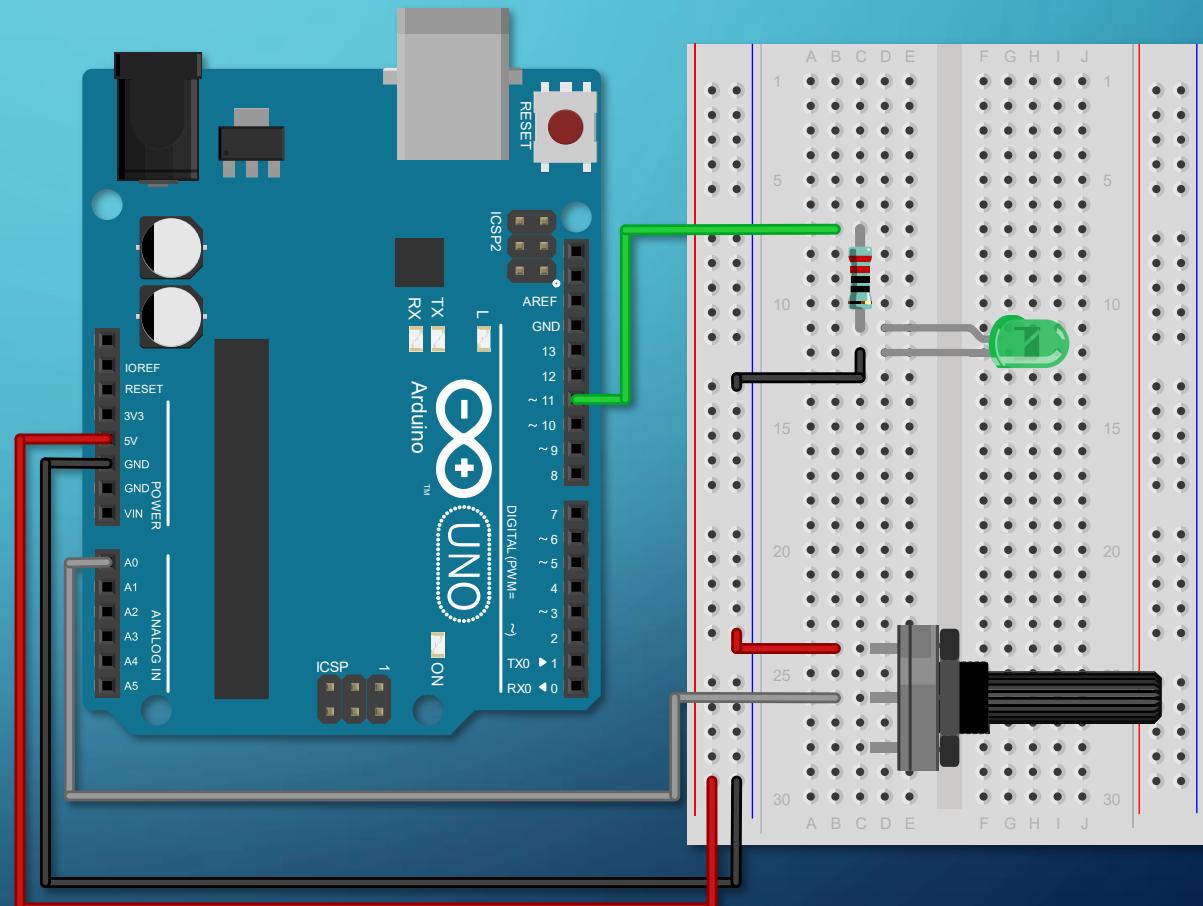
ANALOG PÅ ARDUINO

- Arduino kan også måle analog data
 - Gøres på dens analog input pins
- Den kan også lave et "semi" analogt signal
 - Gøres vha. PWM (Pulse Width Modulation)



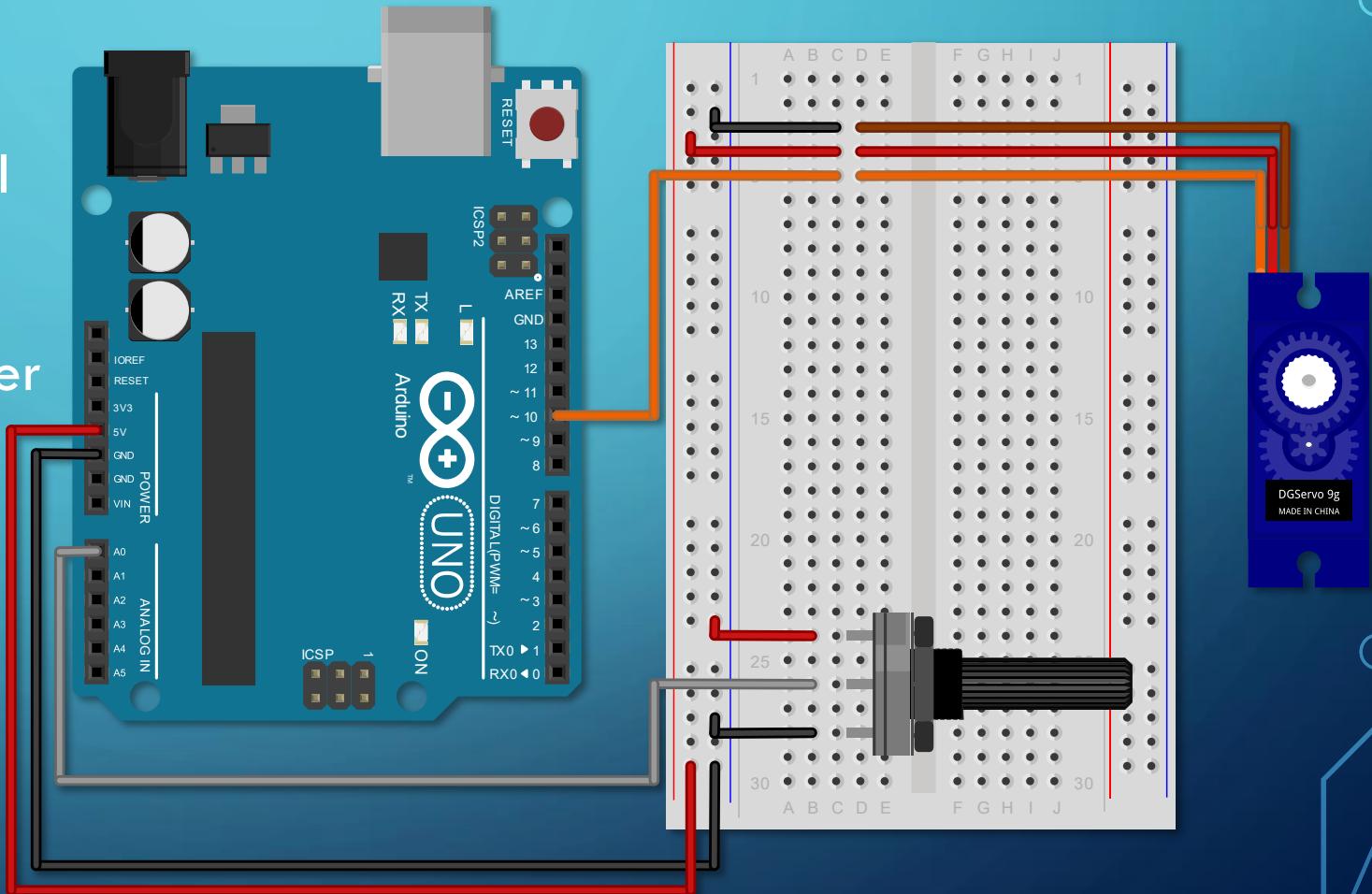
EKSEMPEL 5: ANALOG LED

- Vi aflæser et potentiometer
- Bruger dataene fra den til at indstille en LED
- Status skrives over Serial



EKSEMPEL 6: SERVO

- Vi inkluderer et "library" til at styre en servo motor
- Den styres fra potentiometer
- Status skrives over Serial



ANDRE KOMPONENTER - SENSORER

- Temperatur-/fugtighedssensor
- Accelerometer/gyroskop
- Ultralyds afstandssensor
- Kommunikerer oftest gennem protokoller
 - Håndteres af libraries



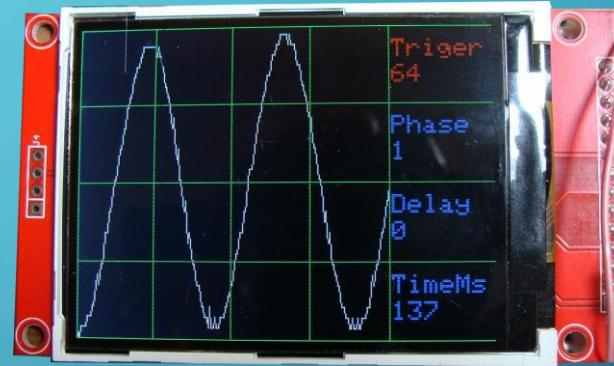
ANDRE KOMPONENTER - AKTUATORER

- Stepper Motorer
- Højtalere
- Lineære aktuatorer
- Stempler
- De fleste bruger ingen, eller simple protokoller
 - Stepper og højtalere har libraries



ANDRE KOMPONENTER - DISPLAYS

- Tekstdisplays (LCD)
- OLED
- TFT LCD
- Mange forskellige libraries for hver
 - Afhængigt af formål





An Arduino Uno R3 microcontroller board is shown from a top-down perspective, tilted slightly. A silver ESP32 module is mounted on top of the board. The board features a blue PCB with various electronic components, including a central microcontroller chip, capacitors, and resistors. The digital pins are labeled along the right edge, and the analog pins are labeled along the top edge. The text "ARDUINO.CC" is printed on the bottom right corner of the board. A large, semi-transparent black rectangular box is centered over the board, containing the white text "SETUP AF ESP32".

SETUP AF ESP32

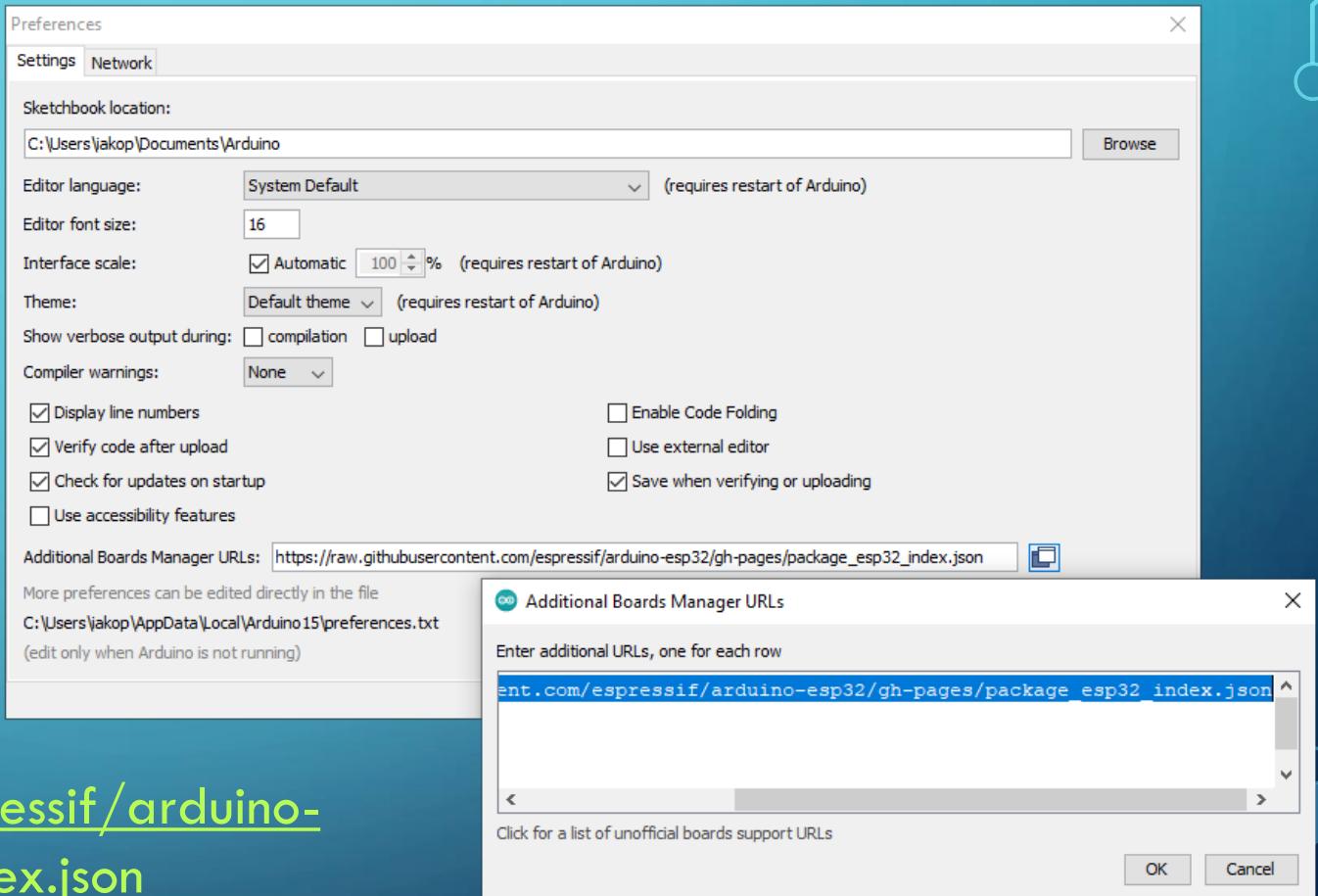
ESP32

- WiFi SoC
 - "System on Chip"
 - Læs: MCU med indbygget WiFi
- Arvtager for ESP8266
- Kan programmeres gennem Arduino IDE
- Kræver lidt setup



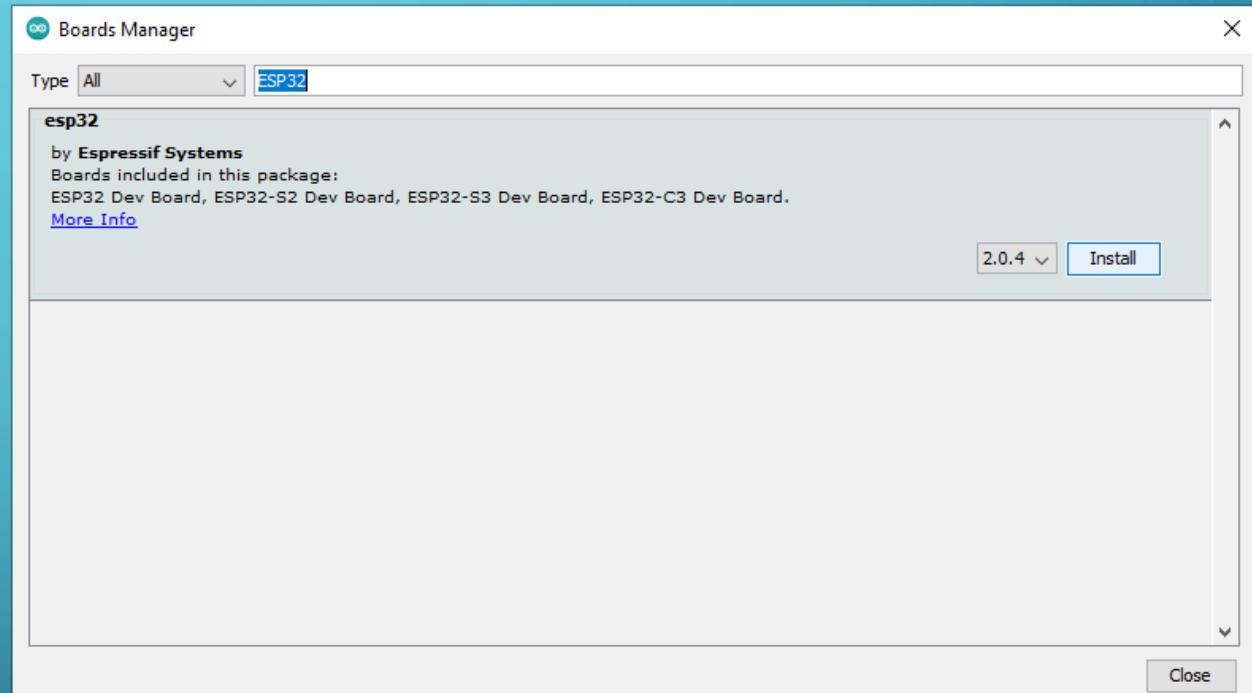
BOARD DEFINITION

- I Arduino IDE:
 - "File > Preferences"
- Additional board manager URLs:
- Indsæt:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



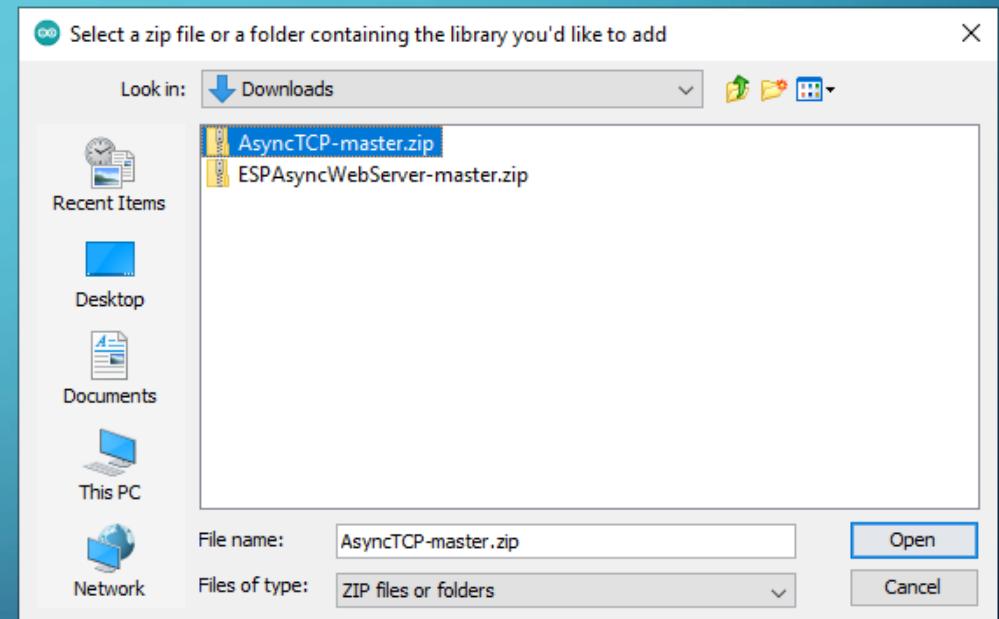
BOARD DEFINITION

- I "Tools > Board > Board Manager"
- Søg efter ESP32 og tryk Install
 - Tager et stykke tid



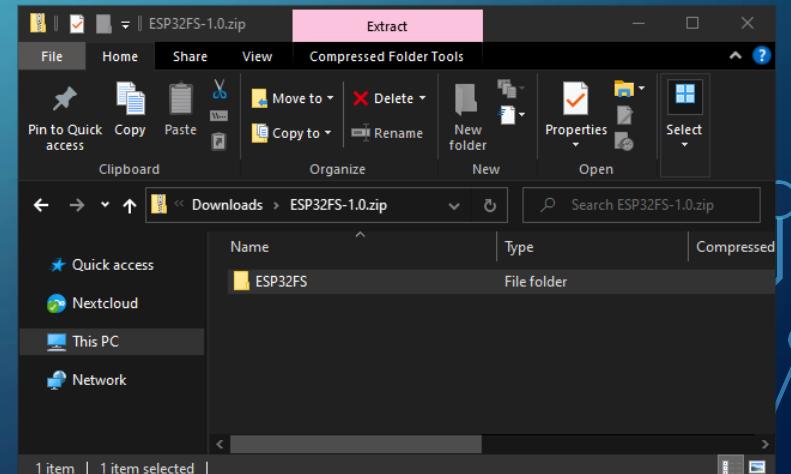
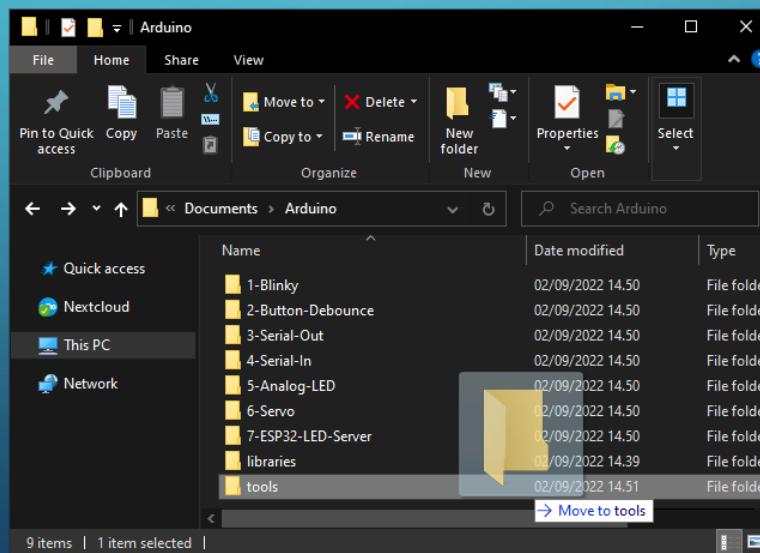
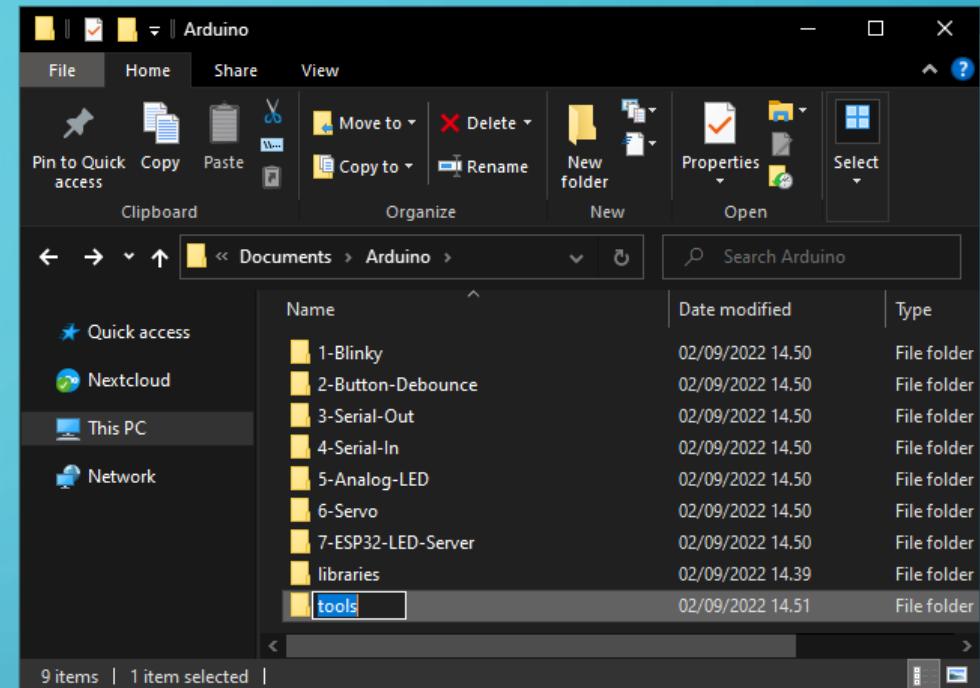
EKSTERNE LIBRARIES

- Følgende to links skal downloades:
- <https://github.com/me-no-dev/ESPAsyncWebServer/archive/refs/heads/master.zip>
- <https://github.com/me-no-dev/AsyncTCP/archive/refs/heads/master.zip>
- I Arduino IDE "Sketch > Include Library > Add .ZIP Library..."
 - For hvert library, installer det



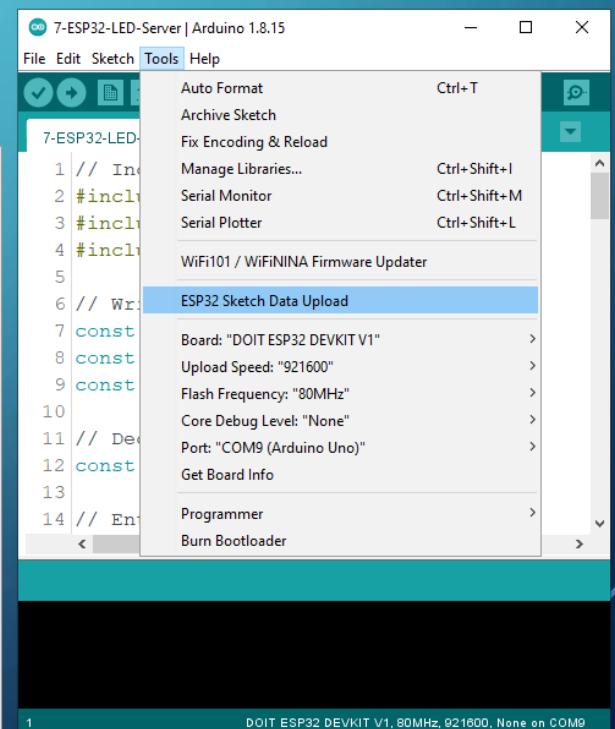
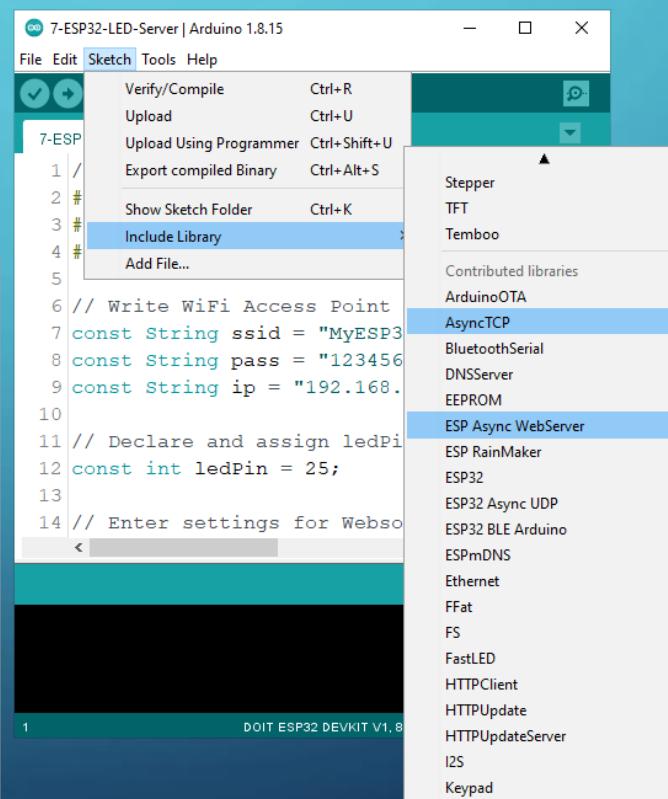
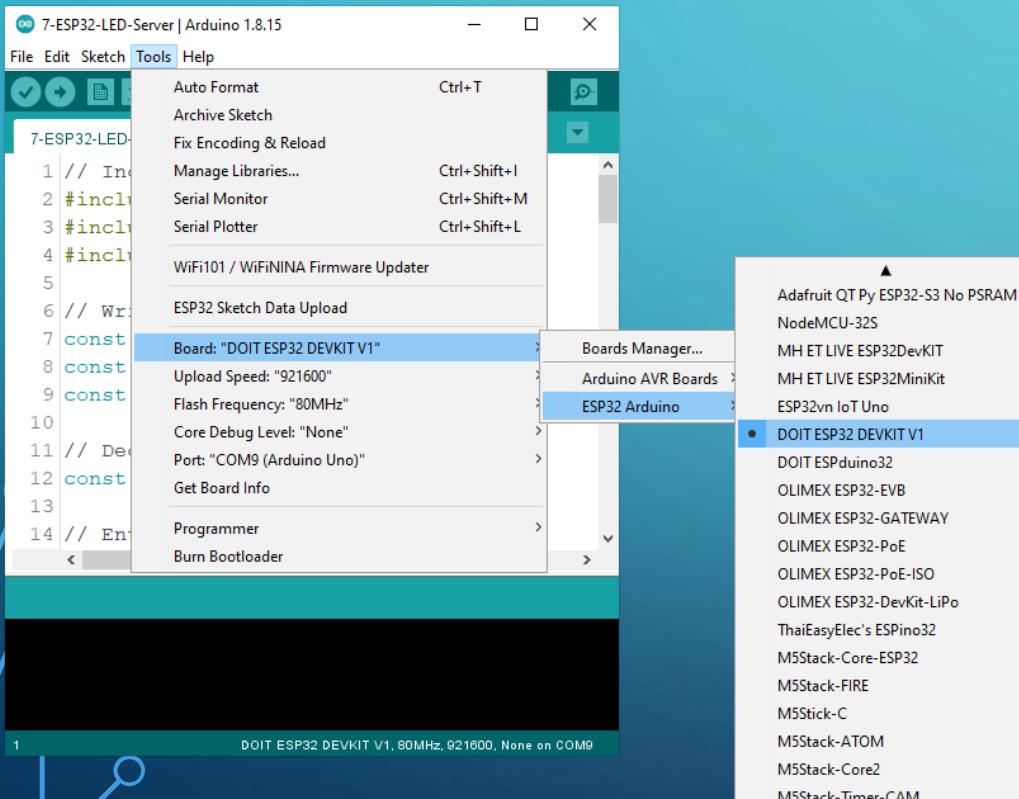
EKSTERNE TOOLS

- Download dette værktøjsplugin:
- <https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/download/1.0/ESP32FS-1.0.zip>
- I din lokale "Documents/Arduino" folder lav en mappe "tools"
- Pak ESP32FS ud i tools mappen
- Genstart Arduino IDE



EKSTERNE TOOLS OG LIBRARIES

- Vi skal nu kunne vælge disse fra Arduino IDE





IOT PÅ ESP32



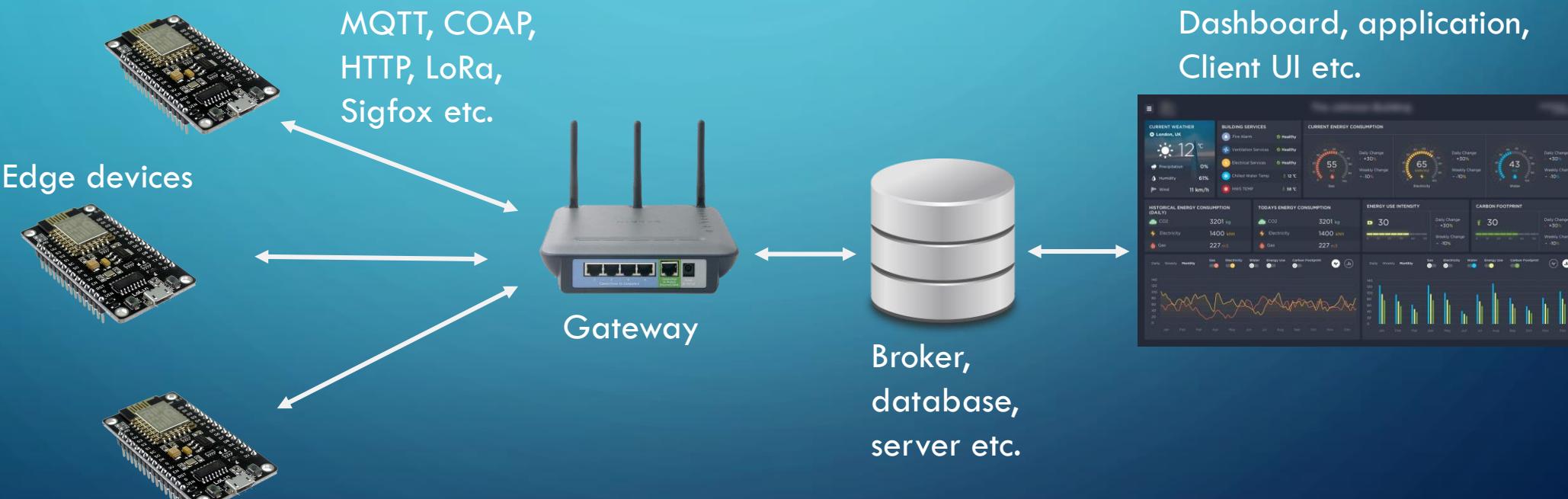
IOT GENERELT

- IoT (Internet of Things) dækker over genstande der er "tilgængelige" over netværk
- Hjemmenetværk:
 - Home assistants
 - Philips Hue
- Ude i felten:
 - WasteHero
 - TrapMe
 - MinkPolice



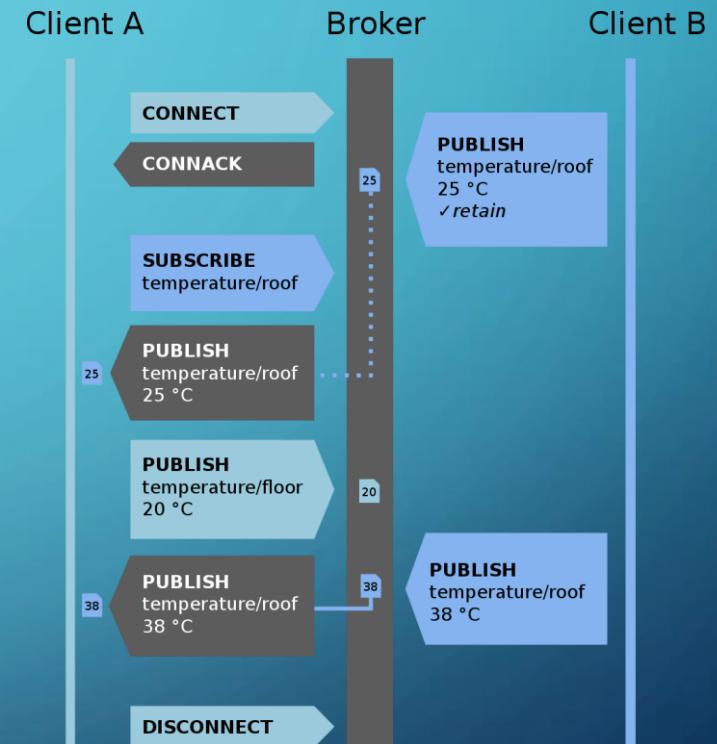
IOT GENERELT

- Opkobling sker ofte ud fra denne form:



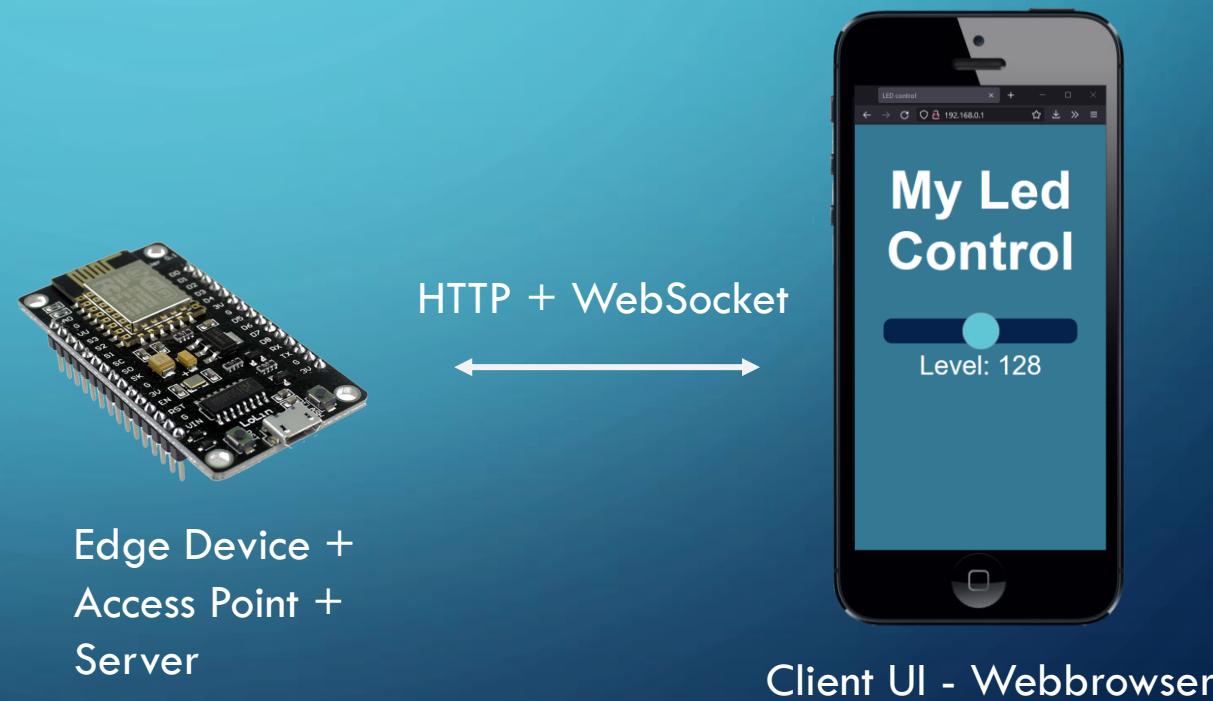
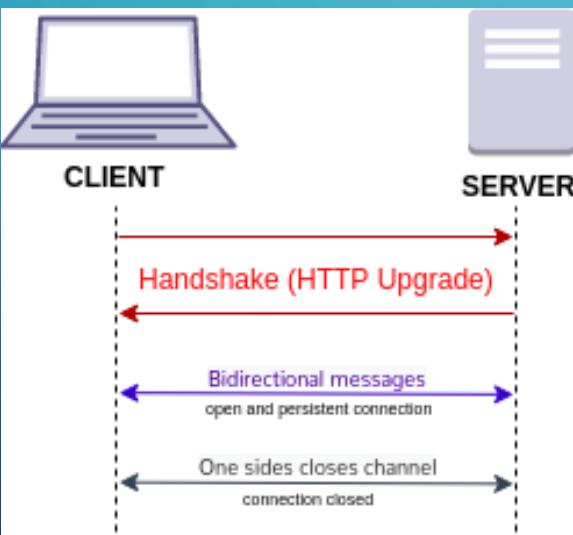
IOT GENERELT

- En af de mest udbredte multi-device protokoller er MQTT
 - Den er simpel og hurtig
- Kræver klienter og en broker
 - Eks. Fra wikipedia
 - To termostater på netværket snakker sammen med broker
 - Udveksler data asyntkront, broker offentliggør data som er relevant



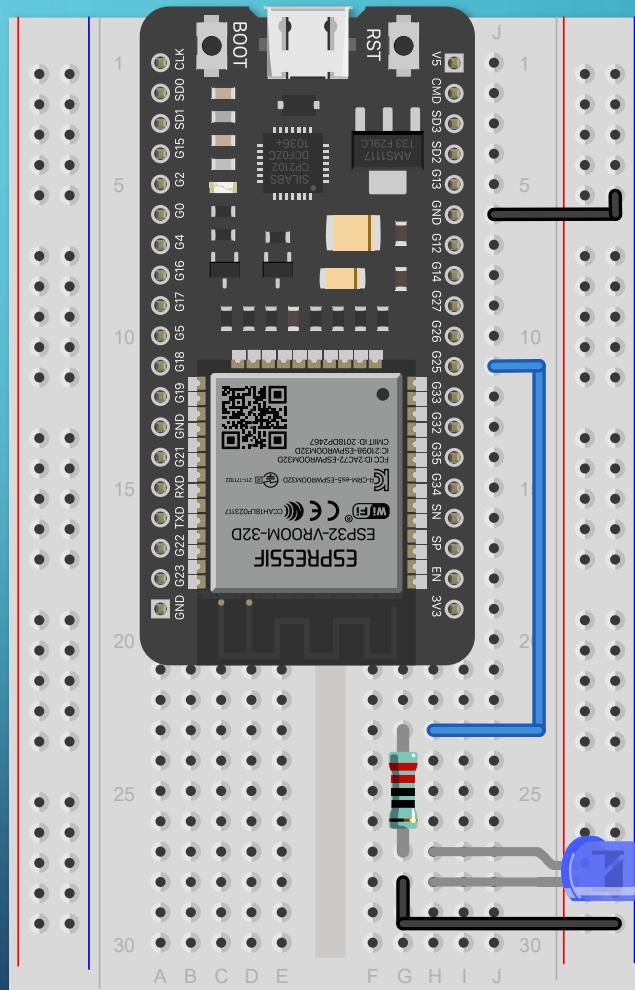
ESP32 WEB SERVER

- For at undgå for meget netværksopsætning forbinder vi direkte til ESP32
- Websocket er bare direkte beskeder "a-la Serial"



EKSEMPEL 7: ESP32 LED SERVER

- Vi bygger blot en LED til ESP32
- Det meste af tiden bruges til at kode, dekorere og opsætte siden





AFRUNDING

OPSAMLING

- Nu kender I til de basale principper
 - Digital I/O
 - Serial UART
 - Analog I/O
 - Introduceret til diverse andre komponenter
- Har sat ESP32 op til Arduino udvikling
- Kan bygge en simpel webservice
 - ESP32 som server og access point
 - HTML/Javascript til kommunikation
- Introduceret til videregående IoT koncepter
 - MQTT
 - COAP

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/ArduinoCrashcourseloT>
 - Arduino IDE Download
 - <https://www.arduino.cc/en/software>
 - Arduino Language Reference
 - <https://www.arduino.cc/reference/en/>